AMSTRAD

PROGRAMMES D'INTELLIGENCE ARTIFICIELLE EN BASIC

DANIEL LEBIGRE

SYBEX

AMSTRAD

PROGRAMMES D'INTELLIGENCE ARTIFICIELLE EN BASIC

AMSTRAD

DANS LA MÊME COLLECTION

Amstrad jeux d'action, P. Monsaut
Amstrad premiers programmes, R. Zaks
Amstrad 56 programmes, S.R. Trost
Amstrad guide du BASIC et de l'AMSDOS, J.-L. Gréco/M. Laurent
Amstrad exploré, J. Braga
Amstrad programmation en assembleur, G. Fagot-Barraly
Amstrad guide du graphisme, J. Wynford
Amstrad CP/M 2.2, A. d'Hardancourt
Amstrad astrologie, numérologie, biorythmes, P. Bourgault
Amstrad graphisme en trois dimensions, T. Lachand-Robert
Amstrad Multiplan, Amstrad
Amstrad CP/M plus, A. d'Hardancourt
Amstrad Astrocalc, G. Blanc/P. Destrebecq
Amstrad gagnez aux courses, J.-C. Despoine
Amstrad créer de nouvelles instructions, J.-C. Despoine
Amstrad Locoscript, B. Le Dû
Amstrad mise au point des programmes BASIC, C. Vivier/Y. Jacob
Amstrad jeux en assembleur, E. Ravis
Amstrad techniques de programmation des jeux, G. Fagot-Barraly
Amstrad routines en assembleur, J.-C. Despoine
Amstrad mieux programmer en assembleur, T. Lachand-Robert
Amstrad jeux de réflexion, G. Fagot-Barraly
Amstrad programmes en langage machine, S. Webb (*à paraître*)
Amstrad guide du Logo, A. d'Hardancourt (*à paraître*)
Amstrad introduction à la programmation en assembleur du Z80, A. d'Hardancourt (*à paraître*)
Amstrad guide du DOS, Amstrad (*à paraître*)
Amstrad systèmes d'exploitation, Amstrad (*à paraître*)
Amstrad gagnez à la Bourse, J.-C. Despoine (*à paraître*)
Amstrad Startext (logiciel de traitement de texte) (*à paraître*)
Amstrad PCW 8256/8512 guide du BASIC et de Jetsam, J.-L. Gréco/M. Laurent (*à paraître*)
Amstrad programmez votre traitement de texte, J.-C. Despoine (*à paraître*)

DANIEL LEBIGRE

AMSTRAD

PROGRAMMES D'INTELLIGENCE ARTIFICIELLE EN BASIC

Paris • Berkeley • Düsseldorf

Sybex n'est lié à aucun constructeur.

Tous les efforts ont été faits pour fournir dans ce livre une information complète et exacte. Néanmoins, Sybex n'assume de responsabilités ni pour son utilisation, ni pour les contrefaçons de brevets ou atteintes aux droits de tierces personnes qui pourraient résulter de cette utilisation.

Copyright © Sybex, 1986.

Tous droits réservés. Toute reproduction même partielle, par quelque procédé que ce soit, est interdite sans autorisation préalable. Une copie par xérographie, photographie, film, bande magnétique ou autre, constitue une contrefaçon passible des peines prévues par la loi sur la protection des droits d'auteur.

ISBN 2-7361-0220-7

SOMMAIRE

Introduction 9

I. Invitation au voyage au pays de l'intelligence artificielle 13

II. Prouver le mouvement en marchant 19

III. Quelques idées reçues naturelles sur l'intelligence artificielle 23

IV. Que la lumière soit 33

Étape E1. *L'inertie aveugle* 36
Étape E2. *Des obstacles inutiles* 41
Étape E3. *Un premier clin d'œil* 45
Étape E4. *Les heuristiques* 47
Étape E5. *Un environnement variable* 53
Étape E6. *Une heuristique de survie* 57
Étape E7. *Vivent les robots!* 62

V. À la recherche d'une martingale 65

Étape E8. *Un peu d'ordre* 67
Étape E9. *La puissance des prédicats* 71
Étape E10. *Roméo passe l'impasse* 77
Étape E11. *Roméo se tait* 80
Étape E12. *Plus de compétence, moins de liberté* 84
Étape E13. *Les fonctions d'évaluation* 89
Étape E14. *Silence on tourne* 95
Étape E15. *De la vigueur dans la tactique, du flou dans la stratégie* 99

VI. Concerto et variations pour deux robots 105

Étape E16. *Toi et moi* 107
Étape E17. *Une heuristique pipée pour un pas de deux* 111
Étape E18. *Diversité qualitative. Les fonctions logiques* 115
Étape E19. *Des robots qui broient du noir. Le test de Turing* 120
Étape E20. *Le fil d'Ariane* 124

VII. Le passé au service du présent. L'apprentissage 129
Étape E21. *Séparer le bon grain* 131
Étape E22. *Ne pas persister dans l'erreur* 136
Étape E23. *Devancer l'erreur* 140
Étape E24. *Un curieux contrat d'apprentissage. Dimension sociale de l'intelligence* 145
VIII. Généralisation de l'utilisation des fonctions logiques 153
Étape E25. *La puissance simplificatrice* 155
Étape E26. *Cinq sens pour un dangereux itinéraire* 160
IX. L'avenir au service du présent. L'anticipation 169
Étape E27. *Une anticipation de navigation... et un concours proposé au lecteur* 171
Étape E28. *Une anticipation de dégagement à profondeur 1* 176
Étape E29. *Des aller et retour à blanc* 184
X. Quelques parties de bras de fer en perspective 191
Étape E30. *Course-poursuite au cirque* 196
Étape E31. *Une piste peu carrossable* 201
Étape E32. *Sus au méchant* 205
Étape E33. *Banc d'essai pour heuristiques concurrentes* 210
Étape E34. *Qui dit mieux?* 214
Étape E35. *Une question finale insidieuse* 218
Conclusion 221
Annexe 1. Tableau des heuristiques utilisées par les robots dans le cours du livre 223
Annexe 2. Répertoire de termes clés utilisés 224
Annexe 3. Formulation générale des heuristiques du robot 225
Bibliographie 226

INTRODUCTION

"La plupart des occasions de troubles du monde sont grammairiennes."
M. de MONTAIGNE.

"Automates pensants, mus par des mains divines."
VOLTAIRE.

Sur l'écran d'un micro-ordinateur quelconque du commerce, doté d'un ordinaire BASIC [1], on peut, par simulation, doter des objets mobiles du sens de la vue, de l'ouïe,... de l'odorat, et leur insuffler un comportement surprenant, dans des situations imprévues, les hissant ainsi au niveau de *robots* doués d'*Intelligence Artificielle* (I.A.)

La formule magique ainsi tombée, il faut se ressaisir, redescendre des Mille et Une Nuits vers les petits matins terrestres, revenir aux sources et s'interroger sur les mots, ces incontournables périphériques d'entrée et de sortie de nos idées.

Prise littéralement donc, l'intelligence artificielle serait en quelque sorte une entreprise de contrefaçon d'un bien déjà surabondamment répandu sur la Terre et dont, de surcroît, on ignore la nature.

Les faux-monnayeurs s'esbaudiraient d'un tel dessein : fabriquer des billets inimitables que tout le monde possède, risque immense et profit nul.

Rares sont les inventions verbales aussi ambiguës que cette expression, et qui aient un tel succès...

C'est qu'en psalmodiant cette superbe formule, le mortel ordinaire se hisse au niveau du génial sorcier qui la créa et entre ainsi gratuitement dans son palais ; mais ce n'est alors qu'un palais des mirages, irréel et désert.

Car "il est plus facile d'élever un temple que d'y faire descendre l'objet du culte [2]".

Miracle pourtant, cette chose-là existe mais on ne sait exactement ce qu'elle est, et elle n'a aujourd'hui trouvé ni son vrai nom ni sa demeure.

---

1. Les instructions les plus sophistiquées dont on aura besoin concernent le générateur de nombres aléatoires (RND) et les variables indicées à deux dimensions (DIM, f(m,n)) dont sont dotés tous les micro-ordinateurs, même en gamme moyenne. Sur moniteur monochrome, les couleurs choisies donnent une gamme de tons gris-vert.

2. Samuel Beckett, *L'Innommable*, Éditions de Minuit.

Les précieuses pépites en sont disséminées et il faut marcher, monter, revenir, tomber puis creuser pour en recueillir parfois quelqu'une. Le pionnier la regarde alors, perplexe, car il ne sait si elle est vraie ; d'autres pierres vulgaires lancent souvent des reflets aussi brillants, et même les experts s'y trompent.

Le trésor gonfle cependant au soir de chaque étape, et demain l'on saura de mieux en mieux identifier les traits de l'inconnue lorsque les multiples colorations que les prismes actuels lui donnent en auront révélé l'incandescente lumière. Le moment viendra alors de lui donner en baptême son nom véritable, plutôt qu'un faux titre de gloire affublé d'un faux blason. Il faudra aussi que ce nom soit beau à entendre ; il faudra le tenir à l'abri des habituels et insupportables suffixes de la manitique, car cette chose-là sera faite de la substantifique moelle humaine, et non des apparents états d'âme d'une machine.

Ce matin, adoptons le pseudonyme universel et partons joyeusement en quête de notre Arlésienne. L'aventure est maintenant plus facile car les chemins sont déjà défrichés, des escaliers aménagés dans le roc, des sites exploités, des lingots coulés à partir de fragments extraits d'inépuisables mines. Le commerce commence à s'en emparer, attestant la valeur de la chose en "l'élevant au statut de marchandise [3]" !

Les pionniers marchent devant nous, parfois démunis mais toujours enthousiastes. Suivons-les, car ils sont dignes de l'admiration due à cette autre belle inconnue, l'intelligence naturelle, dont la majesté se passe de majuscules. Peut-être un jour serons-nous capables nous aussi de donner quelque coup de piolet, ajoutant un rai d'or à la naissante aurore.

Le thème choisi pour cette incursion au pays de l'I.A., celui de petits bonshommes déambulants, tend à en faire plutôt une agréable excursion, en donnant aux concepts abstraits souvent utilisés en ce domaine des visages familiers et accessibles qui, loin d'affaiblir leur portée théorique, la fortifient au contraire au contact d'une réalité tangible, simplifiée mais non déformée.

Et puis cela change quelque peu des sujets habituels d'initiation, taquin, tic-tac-toe, reconnaissance de lettres, construction de systèmes experts rudimentaires, contre lesquels on ne peut formuler qu'un reproche : ils se fondent sur des exemples déjà abstraits pour éclairer des abstractions. Les robots baladeurs, quant à eux, sont en chair et en os, au premier degré du concret, comme seule le permet une merveille aussi abstraite que l'image vidéo...

Je souhaite au lecteur, au fil des étapes, la même jubilation intérieure que celle ressentie lors de la création des minuscules personnages de ce

3. "Les intelligences artificielles", *Sciences & Avenir*, n° spécial 1985.

livre, avec leur univers rocambolesque, leurs perceptions de Petits Poucets, leur ténacité, l'optimisme surréaliste qui semble émaner de leur comportement et leur air de dire, provocateurs : "L'homme de génie est celui qui m'en donne[4]."

Mais le plus surprenant dans la fréquentation des robots, c'est qu'ils finissent par nous donner de si curieuses idées sur les hommes...

4. Paul Valéry, *Mauvaises Pensées et Autres*, Gallimard.

I

INVITATION AU VOYAGE AU PAYS DE L'INTELLIGENCE ARTIFICIELLE

"... le feu de ses prunelles pâles qui me contemplent fixement..."

C. BAUDELAIRE.

Attention, terrain piégé, informatique-alibi, de drôles d'idées camouflées sous le riant gazon du BASIC, autant prévenir.

Au seuil de cet étrange voyage, il est nécessaire de préciser que la substance du livre est directement accessible par son texte, indépendamment du cortège des programmes. Ces derniers n'ont qu'un rôle complémentaire, celui d'agents doubles dans des services très spéciaux. D'abord, ils furent les embryons de l'ouvrage, des supports de réflexion et d'errance, des tremplins du possible et des témoins de l'impossible lorsque les limites de la machine, de sa langue et de son manipulateur s'avéraient atteintes.

Ensuite, ces petits programmes auront une vocation démonstrative pour le cas où le lecteur ne croirait pas sur parole aux extravagants récits qu'on lui conte et, comme saint Thomas, préférerait y regarder lui-même de plus près.

Enfin, et ce sera le troisième rôle des agents doubles, ils permettront d'exorciser le démon, celui du caractère prétendument artificiel de l'intelligence mise en œuvre, et de constater que les petites créatures qui se promènent sur l'écran sont comme nous des joueurs impénitents, on ne peut plus naturels.

Le seul moyen d'assister effectivement à ces réjouissantes déambulations consiste à les concrétiser en programmes, mais cela sera facile lorsqu'on aura d'abord saisi l'essence, en langage clair, de ce qu'ils expriment en langage codé.

Pour cette approche limitée mais vraie de l'Intelligence Artificielle, au sens [1] qu'on lui donne aujourd'hui, la langue naturelle est la mieux capable de servir de premier guide, surtout la nôtre avec ses inépuisables ressources de clarté, de finesse, de souplesse et de puissance, son irremplaçable aptitude à exprimer les raisonnements formels ou les jeux informels de l'esprit.

Il faut en effet savoir que les langages actuels d'ordinateurs, même les plus évolués, les mieux structurés, les mieux adaptés — LISP, PROLOG,... —, sont encore, malgré les immenses progrès qu'ils reflètent, pauvres et rigides et qu'ils requièrent un maniement rigoureux, au moins aussi artificiel que l'intelligence qu'on veut leur faire traduire.

Ces défauts qui, n'en doutons pas, s'atténueront [2], concernent d'une part la *représentation* du monde réel sous des formes symboliques exploitables par un ordinateur : informations quantitatives et *qualitatives*, données, règles, "culture", c'est-à-dire tous les *supports* de la pensée, et d'autre part le *traitement* de ces représentations, c'est-à-dire *les moteurs*

1. On devrait dire "aux sens" car, en ce domaine nouveau, on en est encore au stade du jaillissement et de la dispersion des idées.

2. E. Feigenbaum et P. McCorduck, *La Cinquième Génération*, InterÉditions.

de la pensée : manipulation des connaissances, tri, classement, combinaison, sélection, inférences ; ces deux opérations, représentation et traitement, permettant en fin de compte de *maîtriser une situation*, finalité de tout acte intelligent. Bien entendu, le BASIC se situe lui-même au plus bas dans cette déjà courte échelle des valeurs.

C'est pourquoi, pour traiter d'un sujet aussi délicat, il faut d'abord se servir de ce qui s'est fait de mieux comme langue : la nôtre. Sa beauté et ses pouvoirs sont infinis, tout détail y compte mais tout excès la ternit. Ainsi, l'étrange puissance simplificatrice du minuscule accent aigu, qui transfigure la musique et la signification des mots : l'on piège, mais on est piégé...

Ce sont ses qualités mêmes qui la rendent redoutable, involontairement par facilité ou sciemment par calcul. Mais, utilisée selon sa vraie vocation de clarté, "elle est la seule qui ait une probité attachée à son génie [3]".

Après cette approche naturelle, il sera bien facile, comme déjà dit, de se rabattre sur le BASIC pour concrétiser sur un écran, dans leur simplicité voulue, les exemples proposés. Il suffira, si ce n'est déjà fait, d'apprendre un vocabulaire d'une centaine de mots, articulés sur une syntaxe limpide. Leur consonance anglo-saxonne n'est pas gênante et apporte du *look*... Il n'y a pas de conjugaison, l'ordinateur se contentant de déclarations non équivoques du genre MAITRE = 0 ou ROBOT = 1, et du présent de l'impératif. À la limite, il tolère des conditionnels aussi évolués que SI FROID ET CIEL GRIS ALORS MOI PAS SORTIR. Bien sûr, derrière ces formes reposantes, il y a des significations précises : on ne parle pas en BASIC pour ne rien dire, et ceci compense en bonne partie cela. Enfin, on y donne cyniquement des ordres puis des contrordres ; le dernier est toujours le bon, du genre : MAINTENANT A = A + 1.

Mais les pensées les plus subtiles, les théorèmes les plus inattendus sont toujours à la dernière marche d'un escalier d'évidences.

Un savant génial utilise bien, une fois par siècle, ce qui ressemble à un ascenseur pour ses contemporains, mais cela ne s'en avère pas moins, après coup, toujours une simple marche.

L'ordinateur, avec ses jeux d'instructions, en détient un stock complet à la disposition des programmeurs auxquels il ne reste plus qu'à bâtir de superbes escalators.

Quel adulte responsable n'a pas ainsi accepté à une époque de sa vie, pour communiquer avec ses enfants, de transformer et de simplifier pro-

---

3. Antoine de Rivarol, *Discours sur l'universalité de la langue française.*

visoirement son vocabulaire, à seule fin de faire ensemble un bout de chemin ?

Qui ne s'est alors aperçu que ce sous-langage dépouillé contenait en fait l'essentiel et permettait ensuite l'élévation progressive des mots, des échanges, et de l'intelligence qu'ils sécrètent ?

Pour en terminer avec ces questions de langage, comment distinguer par *leur forme*, un programme d'Intelligence Artificielle et un programme informatique ''classique'' ?

En effet, sans entrer dans les détails techniques, rien ne ressemble plus à un programme en BASIC qu'un autre programme en BASIC : il y a des lignes numérotées, quelques remarques en clair, et des litanies d'instructions. Pourtant, en y regardant de plus près, on relèvera dans les programmes de ce livre, tout comme dans les explications et commentaires qui en sont donnés, un certain nombre de particularités qui reflètent le caractère du sujet exposé.

D'abord, l'intervention à une fréquence inhabituelle de variables[4] affublées de noms évocateurs, OBS, ESS, ATT et autre PIL qui symbolisent et représentent, non pas des *quantités*, mais des *propositions qualitatives vraies ou fausses* telles que : IL Y A UN OBSTACLE. Certaines de ces variables, reflétant par exemple le rang d'une position ou l'état d'un point de l'écran, seront dotées d'un ou deux indices ou *arguments*[5] et passeront ainsi au statut philosophique supérieur de *prédicat* dont nous verrons, à l'aide des exemples, la simple signification.

Enfin, dans certains programmes, ces prédicats pourront prendre des valeurs quelconques et non plus seulement les deux valeurs vrai ou faux. Ils deviennent alors de véritables *fonctions logiques*, celles que nous utilisons inconsciemment pour débroussailler nos problèmes quotidiens. Leur puissance de représentation et de manipulation est considérable. En fait, sans variables indicées, le BASIC serait pratiquement inapte à toute programmation d'I.A.

Ensuite, un usage intensif du générateur de nombres pseudo-aléatoires. En clair, on joue souvent à la roulette. C'est que, dans la vie des robots, les données, les obstacles, l'environnement doivent être peu ou prou distribués au hasard, sinon à quoi leur servirait-il d'être devenus intelligents ?

Il y a un autre motif moins évident mais essentiel à ces habitudes de casino. Le monde réel présente souvent des données incomplètes, floues, voire contradictoires. André Gide note qu'il entre dans toutes les actions

---

4. Variables numériques ou variables chaînes de caractères.

5. Indices ou arguments uniquement numériques permis par le BASIC, mais qui peuvent très bien correspondre aussi à des faits qualitatifs ; l'autre avantage décisif des indices réside dans leur pouvoir de mémorisation du déroulement d'un scénario et, en conséquence, dans leur pouvoir d'expression de la notion d'*apprentissage*.

humaines plus de hasard que de décision et Paul Valéry en donne le saisissant contrepoint : que de choses il faut ignorer pour agir... En quelque sorte, on ne sait pas tout ce qu'il faudrait savoir pour bien faire, mais si l'on savait tout l'on ne ferait pas.

Heureusement, cette nécessité de fonder partiellement nos actes sur le néant — le hasard n'est-il pas le néant qui se manifeste ? — s'avère souvent aussi efficace que la part raisonnée du comportement, à condition de rester dans certaines limites. Car il est fréquent que le problème à résoudre pardonne une erreur : ou bien la solution trouvée est acceptable, même si elle n'est pas la meilleure, ou bien l'on peut reprendre ses billes et recommencer la partie.

En outre, ces aléas donnent curieusement une marge de manœuvre, des degrés supplémentaires de liberté, et confèrent au déroulement de l'action son caractère varié, indéterministe, inattendu, qui en fait le sel et constitue aussi, non seulement la raison d'être, mais la substance de l'*apprentissage*.

Enfin, on butera à chaque détour sur les pointes d'aiguillage des décisions conditionnelles, les fameux IF THEN (si alors), car les robots, comme chacun sait, y regardent à deux fois avant de faire un pas.

En fait, d'autres particularités apparaîtront, dont les motifs ne seront clairement saisissables qu'à l'appui d'exemples précis.

Voilà pour la forme. En ce qui concerne le fond on a en général affaire, dans les programmes ordinaires de simulation de mouvements, à des projectiles, à des boules de billard, à des astéroïdes, au mieux à des fusées, c'est-à-dire à des corps mobiles plus ou moins *inertes* qui, une fois lancés, obéissent aveuglément à leur destin déterministe. La multiplicité des trajectoires provient uniquement de la variété possible infinie des ordres de lancement : direction et vitesse initiales.

Rien de tel ici. On verra peu à peu émerger de la brume inertielle, d'abord quelques lueurs, puis un faisceau de plus en plus riche de comportements cohérents, une espèce d'autonomie, une recherche progressive du meilleur moyen d'atteindre le but assigné et de déjouer les contraintes et les obstacles.

Au fil des premiers programmes, le *projectile* deviendra *missile*, puis *robot*. La solitude du programmeur se peuplera d'une étrange compagnie car il y aura bientôt sur l'écran, non plus *quelque chose qui circule*, mais *quelqu'un qui veut aller quelque part*...

II

PROUVER LE MOUVEMENT EN MARCHANT

"L'homme est ainsi bâti : quand un sujet
l'enflamme, l'impossibilité disparaît à son âme."
J. DE LA FONTAINE.

Quelques mots pour développer les raisons du choix du thème des simulations de mouvements dans un ouvrage présentant des programmes d'initiation à l'I.A., raisons déjà effleurées dans l'Introduction.

Chacun possède au moins un domaine dans lequel il excelle et à travers lequel il sait s'exprimer pleinement. Ce peut être aussi bien la pêche au lancer, le calcul mental, le bridge, le tir à la carabine, le jeu vidéo de Pac-Man, les paris hippiques, l'expression verbale, l'aéromodélisme, l'art de séduire, les problèmes de géométrie, le motocross, la danse, la menuiserie, la cuisine, l'astronomie, la musique...

Ces domaines privilégiés possèdent trois caractéristiques communes.

D'abord, ils peuvent concerner un éventail très large d'activités et, fait essentiel, aussi bien celles réputées manuelles ou matérielles que celles noblement cataloguées comme intellectuelles ou spirituelles. Il est aisé de constater qu'en fait cette tentative de mise en deux bocaux séparés est dénuée de sens.

Ensuite, les intéressés peuvent y atteindre effectivement à un haut degré de perfection de la connaissance, de l'intelligence des situations, hors de portée du premier venu, l'amateur dépassant alors parfois le professionnel. Ils constituent en quelque sorte l'exaltation, dans un compartiment précis, des facultés individuelles, et donc la preuve que celles-ci existent à l'état potentiel, fleurissant dès que certaines conditions sont remplies.

Enfin, ils deviennent toujours l'objet d'une véritable passion.

Dans son livre *Jaillissement de l'esprit* [1], Seymour Papert lie cette germination aux "matériaux" dont l'individu a disposé, en particulier dans son enfance, et avec lesquels il a eu l'occasion de jouer et de penser, de se construire un monde de connaissances progressivement enrichies, détaillées, structurées, approfondies, *et aimées*. Pour Papert par exemple, les jeux d'engrenages l'avaient fasciné ; il les avait manipulés et fait tourner dans tous les sens, sous toutes les formes, du moteur à ressort au différentiel de voiture, et avait fini par s'identifier corporellement à eux, tournant avec eux, et les assimilant ainsi par l'intérieur, digérant leurs propriétés les plus subtiles, transformant les engrenages en substance de Seymour Papert, en esprit de Seymour Papert...

Qui n'a pas eu ses jeux d'engrenages, parfois très variés, à divers âges de la vie ? Qui ne désire les retrouver un jour sur son chemin, s'y réincarner et s'envoler vers un sommet de son rêve, libéré des inhibitions et des faux obstacles ?

Dans un domaine comme l'informatique d'amateur, où la programmation apparaît parfois comme une activité abstraite réservée à quelques originaux, un jeu d'engrenages est le bienvenu.

---

1. Éditions Flammarion.

Il me semble que la simulation des mouvements constitue l'un des plus excellents. Un mouvement a toujours quelque chose d'étrange ; il est l'image de la vie, avec ses inflexions, ses hésitations, ses retours, ses palpitations, ses fulgurances.

On a facilement envie de créer des mouvements, de les exécuter, de s'y identifier comme le joueur de pétanque qui accompagne la boule de ses dynamiques contorsions, comme l'enfant qui navigue le long du ruisseau aux côtés de son bateau de papier, qui clignote des yeux devant les feux du carrefour ou qui tourbillonne avec sa toupie.

Ainsi, pour aborder le sphinx de l'informatique par la petite porte naturelle de la programmation sur micro-ordinateur, la simulation des mouvements m'était déjà apparue comme un banc d'essai idéal [2]. Cela permet, non seulement d'obtenir sur l'écran des effets concrets intéressants, parfois spectaculaires, et de recréer certains aspects du monde réel, mais aussi d'exploiter à fond toutes les ressources d'un micro-ordinateur : manipulation de nombres et de chaînes de caractères, calcul logique, graphisme, couleurs, sons, intervention au clavier ou à l'aide des manettes de jeu pendant le déroulement de l'action.

De même, au moment de proposer les exemples plus spéciaux de ce livre, c'est encore vers les mouvements, pour les mêmes raisons que précédemment, qu'il me paraît opportun de tourner les regards.

En outre, nous verrons que ce thème se prête particulièrement bien à la mise en vedette de nombreux critères reconnus actuellement comme spécifiques des programmes d'Intelligence Artificielle, par exemple les notions d'heuristique, d'apprentissage, de variabilité de l'environnement, d'anticipation et de fonction d'évaluation d'une situation, de programmation déclarative, d'indéterminisme, de données conflictuelles, de droit à l'échec, etc. Cela, certes, dans les limites vraies d'un micro-ordinateur, de sa mémoire et de son langage, mais hors des préjugés et des idées reçues relatives à son inaptitude congénitale à ce genre d'exercice, surtout si les idées et procédés correspondants sont, avant toute démonstration, exposés en clair.

En somme, l'Intelligence Artificielle manquant de naturel, l'intelligence naturelle y obvie en recourant à des artifices.

Mais le mieux à cet égard n'est-il pas, pour prouver le mouvement, de marcher ?

---

2. *Dynamique-Cinématique, Méthode pour la programmation des jeux*, Éditions Sybex.

III

QUELQUES IDÉES REÇUES NATURELLES SUR L'INTELLIGENCE ARTIFICIELLE

"La pensée est une activité à laquelle je me livre rarement."

A. EINSTEIN.

L'I.A. a déjà droit de cité au stade industriel dans quelques domaines, et fait l'objet de recherches importantes avec des spécialistes de haut niveau, des ordinateurs puissants d'architecture nouvelle et des langages également nouveaux particulièrement évolués.

Ces trois critères, savants chercheurs, super-ordinateurs, hyperlangages constituent certes des facteurs éminemment favorables au développement des applications futures, de plus en plus complexes, mais ne constituent nullement des conditions *sine qua non* à la réalisation de programmes d'initiation à l'I.A.

Aucun ordinateur, aucun langage, aucun programmeur ne saurait être *a priori* tenu à l'écart d'une activité de programmation quelle qu'elle soit.

Le livre récent de Mike James [1], et d'autres, fournissent des exemples intéressants sur micro-ordinateur dans des domaines divers, mais qu'on retrouve souvent d'un ouvrage à l'autre en raison de leur réputation démonstrative.

Des considérations peu familières au débutant interviennent parfois en I.A. Elles ne sont pas toujours d'ordre théorique ni spécialement difficiles, et c'est souvent leur vocabulaire et le caractère abstrait de leur exposition qui peuvent dérouter. Elles seront abordées le moment venu sur les exemples précis proposés et sous forme aussi communicable que possible. Car c'est bien la communicabilité qui fait le plus défaut dans le cercle encore fermé des chercheurs en I.A., longtemps confinés dans leurs nécessaires et délicates relations internes. Mais cela, même en France, évolue rapidement. Des revues sérieuses publient des articles "grand public" assez clairs ou des numéros spéciaux [2] qui permettent de faire le point et d'enfourcher la noble monture.

Le beau livre d'Alain Bonnet, l'*Intelligence Artificielle, promesses et réalités* [3], ouvre les portes à un haut niveau mais dans un langage simple où les définitions sont clairement données.

La revue internationale *Artificial Intelligence* permet aussi, bien sûr, de se tenir au courant, mais la lecture est plus difficile.

L'essentiel est de ne pas avoir à s'abreuver à des sources qui, pour expliquer du grec, se croiraient obligées de le traduire en hébreu.

"Puissé-je ne me servir que des mots qui servent aux Halles à Paris ! [4]" et puis encore : "Il faut bien citer ce qu'on ne comprend point dans la langue qu'on entend le moins [5]."

A partir de l'étincelle initiale, la suite est plus simple car l'enthou-

1. *Introduction à l'Intelligence Artificielle sur micro-ordinateur*, Éditions Eyrolles.
2. *Sciences & Avenir*, n° spécial 53, 1985. *La Recherche*, n° spécial 170, 1985.
3. InterÉditions.
4. Montaigne, *Essais*.
5. Voltaire, *Micromégas*.

siasme est presque inévitable ; on peut sans difficulté insurmontable apprendre ce qu'est, dans son essence, l'I.A., puis commencer immédiatement à participer à son bouillonnement général. On peut le faire aujourd'hui en amateur mais ce sera moins vrai demain pour une raison très simple : l'I.A. en est encore, au bon sens du terme, au stade du *bricolage* [6].

Il existe beaucoup d'outils plus ou moins au point, conçus par des artisans quelquefois géniaux et dont le savoir-faire, acquis sur le terrain, s'affirme chaque jour. La plupart de ces outils (langages, méthodes, machines, principes quelquefois, découvertes rarement) font avancer l'entreprise, même si quelques-uns d'entre eux apparaissent, provisoirement peut-être, destinés à demeurer dans les stands du Concours Lépine.

De temps à autre, une idée fulgurante jaillit et donne une arme nouvelle pour investir la forteresse et s'attaquer plus efficacement à la chambre des coffres.

Quant aux produits, on peut en voir une kyrielle de prototypes presque toujours habilement présentés, mais les objets à dénicher sont encore mal définis. Certains sont parfaitement consommables et auront plus tard leur place au glorieux musée de l'I.A.

Dans ce Grand-Pressigny de la pensée, il est possible de rester spectateur critique, corrosif ou bienveillant. Mais il est humainement plus naturel d'être fasciné, d'entrer dans l'arène comme un enfant et, saisissant le premier outil à portée, de se frapper sur les doigts, de mieux entrevoir l'ampleur de l'entreprise, ses vertigineux pièges et la difficulté de poursuivre une construction dont l'architecte n'a pas fourni les plans...

Il n'est pas nécessaire d'être intégré dans des équipes de haut niveau pour participer à l'aventure, voire pour apporter sa pierre.

Les communautés, certes, stimulent et bénéficient de moyens inaccessibles à l'amateur isolé, mais tendent aussi à unifier les opinions, à niveler les imaginations débordantes indispensables dans ce genre d'exercice, à élever des murs d'écoles. Il en fut ainsi pour les espèces vivantes, au cours des âges ; le croisement des idées, comme celui des individus d'une grande population, tend à redistribuer les mêmes gènes et à stabiliser l'espèce. C'est aux Galapagos et non sur les grands continents interconnectés que Darwin put avoir la révélation de l'Évolution, d'un îlot à l'autre, là où les ''monstres'' avaient tranquillement pu échapper à la normalisation.

C'est aussi dans le cocon d'un morne bureau de brevets qu'un original échafauda la théorie de la *relativité*, totalement débranché des cou-

6. Peut-être en sera-t-il toujours ainsi car l'utilisation d'une simple boussole suppose que l'on sache où l'on veuille aller.

rants neuroniques extérieurs, mais parfaitement au fait de *ce qu'il fallait trouver*.

En préliminaire aux exploits de nos robots, essayons, parce qu'il est normal de connaître un peu le terrain sur lequel on va s'aventurer, d'approcher l'I.A. par quelques-uns de ses aspects concrets tels qu'ils sont actuellement reconnus par la communauté des chercheurs.

Au départ de toute exploration, il y a l'inconnu, la jungle, le vrac. Voici, sous cette forme, une liste non exhaustive d'applications ou de pistes de recherche de l'I.A., les unes presque parvenues au stade de l'exploitation courante, d'autres au niveau des laboratoires, d'autres encore à l'état de spéculations embryonnaires ou, pire, déjà jetées dans la poubelle des échecs, quitte à émerger à nouveau plus tard sous des formes nouvelles :

a. L'*aide au diagnostic d'une situation*, par exemple dans le domaine médical ou géologique.

b. L'*aide à la décision de traitement d'une situation*, prolongement logique de la précédente démarche.

a et *b* constituent actuellement le prometteur univers des *systèmes experts* dont les premières recherches, dans les années 60, sous l'impulsion parfois mal accueillie d'Américains comme Buchanan ou Feigenbaum, débouchent maintenant sur des applications presque quotidiennes dans de multiples domaines. Il est nécessaire et suffisant (c'est peu de le dire), pour bâtir un système expert, de *soutirer* les connaissances des meilleurs experts humains du domaine considéré [7], de les transférer avec un mode de représentation adéquat dans la *base de connaissances* du système, et de réaliser le *moteur d'inférence*, c'est-à-dire le programme de manipulations mentales (inductions, déductions,...) capable de conduire à l'objectif visé. Doté d'un tel système, l'utilisateur, *qui doit être lui-même compétent dans le domaine concerné*, introduit les données particulières au cas à traiter dans la *base de faits* (par exemple les symptômes qu'il détecte sur un malade) et, après un dialogue plus ou moins long avec la machine au cours duquel des faits ou des explications complémentaires sont éventuellement demandés ou fournis, il obtient un diagnostic ou une présomption de diagnostic et une proposition de traitement.

c. L'*E.I.A.O.*, enseignement intelligent assisté par ordinateur, qui n'a que peu de ressemblance avec les programmes actuellement commercialisés sous le sigle E.A.O.

---

7. Il s'agit de connaissances proprement dites, mais aussi de règles pour les utiliser, des règles sur les règles (métaconnaissances), etc.

d. Les *jeux de l'esprit* : pour les dames, le backgammon et les échecs, c'est déjà la réussite ; pour le poker et le bridge, c'est un banc d'essai prometteur dont la réussite à terme ne peut faire aucun doute.

e. La *reconnaissance des formes*, pour choisir, saisir, transporter et placer un objet.

f. La *compréhension des images*.

g. La *reconnaissance de l'écriture*, dans l'acception sémantique du terme ; c'est réalisé pour l'écriture d'imprimerie, c'est probablement réalisable pour les écritures cursives manuelles.

h. La *reconnaissance de la parole*, malgré la variété des timbres, des intonations... et des significations.

i. La *traduction automatique*, prolongement des deux applications précédentes, qui a fait l'objet dans un passé récent d'échecs retentissants et qui repart sur de nouvelles bases [8]. La commande des robots en est un cas particulier.

j. La *résolution de questions mathématiques difficiles* : recherche d'intégrales, démonstration inédite de théorèmes,...

k. D'une manière générale, toutes les *activités assistées par ordinateur* (XAO).

l. La *robotique*.

C'est ce dernier domaine qui ouvre les perspectives les plus fascinantes, car il est le seul à conduire l'Intelligence Artificielle au bout de sa logique : l'*Action Artificielle*, et à montrer d'une façon éclatante l'unité indissociable de la pensée et du comportement physique.

D'ailleurs, la commande des robots constitue l'un des secteurs les plus difficiles de l'I.A. ; elle constitue pourtant l'exact équivalent de la tâche de nos nerfs moteurs, ceux qui règlent les gestes de ce qu'on appelle avec légèreté le "travail manuel" ; encore une expression qui dénote la méconnaissance totale d'une simple réalité : nos actes sont les seules manifestations et les seules preuves de notre intelligence, ils *sont* notre intelligence.

Il faut bien entendu considérer l'acte au sens large : faire, mais aussi décider, commander, coordonner une action, écrire, peindre, émettre une opinion, conseiller, extérioriser d'une manière quelconque une contribution à la réalisation d'un dessein.

8. *Cf.* Jacques Pitrat, *Textes, Ordinateurs et Compréhension*, Éditions Eyrolles.

C'est aussi ce domaine, sous le modeste égide d'une trilogie amateur-micro-ordinateur-BASIC, et dans le secteur le plus simple, celui de la simulation des déplacements, que le présent livre se propose d'effleurer.

Le lecteur aura peut-être remarqué que la liste d'applications ci-dessus possède une double propriété curieuse : incomplétude et surabondance. Il faudra dans quelque temps en rajouter, mais il faudra aussi en enlever ; nous y reviendrons.

Essayons maintenant de dégager, à partir de cette liste, ce qu'il est convenu d'appeler l'Intelligence Artificielle.

Pour qu'une application puisse prétendre actuellement ressortir au domaine de l'I.A., elle doit posséder, à des degrés significatifs, plusieurs des propriétés énumérées plus loin et naturellement liées aux deux termes de la définition : *intelligence*, *artificielle*.

Nous commencerons par le second, d'un abord apparemment plus facile...

Artificielle

Les systèmes concernés sont faits par des hommes et pour des hommes et font donc appel *dans leur conception* — en amont — et *dans leur utilisation* — en aval — à l'intelligence humaine, et cela à un niveau souvent élevé car les applications correspondantes sont toujours complexes et plus ou moins nouvelles.

C'est *au stade de leur fonctionnement propre* qu'ils prennent leur caractère artificiel : ce sont en effet des machines ou des ensembles de machines qui opèrent, grâce à leur constitution matérielle d'une part et aux connaissances au sens large (savoir, règles) qui leur ont été injectées d'autre part.

Intelligence

Ce terme est, en l'état actuel et probablement pour longtemps, indéfinissable de manière satisfaisante et complète, tout bonnement parce que nous ne savons pas comment est constitué ni comment fonctionne ce que nous nous plaisons à considérer — avec une parfaite suffisance — comme notre plus beau privilège.

Jean d'Alembert disait que la nature de l'homme est un mystère impénétrable à l'homme quand il n'est éclairé que par la raison seule. Doté d'intelligence certes, mais tout juste assez pour s'en servir et sûrement pas autant qu'il en faudrait pour la définir.

Mieux vaudrait s'adresser au Constructeur, mais, probablement trop occupé pour s'arrêter à ces fariboles, il n'a pas jugé nécessaire de fournir

les plans de ses créations. Peut-être le modèle définitif n'est-il même pas au point si l'on en juge par certaines performances.

La *pensée sur la pensée* est un phénomène d'autoréférence et nous savons, depuis que Kurt Gödel a lancé, avec son célèbre théorème d'incomplétude, un coup de pied dans la fourmilière des philosophes, que nous sommes là dans une impasse, l'impasse de notre propre nature.

Douglas Hofstadter nous éblouit et nous donne le vertige de cette invraisemblable situation : "C'est de cet équilibre entre la connaissance et l'ignorance de soi que naît le libre arbitre[9]."

Quoi qu'il en soit, nous essaierons au long du livre de nous esquisser une silhouette de la belle, silhouette plus ou moins valable, mais toujours fondée sur les exemples concrets que nous rencontrerons.

Pour l'instant, nous pouvons simplement recenser certaines des caractéristiques exigées ou admises, explicitement ou non, pour qu'une application ait droit de cité au fabuleux royaume de l'I.A., sous réserve encore qu'elle les possède à un degré suffisant :

a. Le problème à résoudre, la situation à maîtriser sont en général complexes et comportent des données et des règles nombreuses, d'ordre plus qualitatif que quantitatif ;

b. De ce fait, le programme est amené à manipuler des symboles particuliers de représentation du domaine concerné ;

c. Il y a un but final précis ;

d. Il existe un environnement d'obstacles et de difficultés ;

e. Cet environnement varie d'une opération à l'autre ou même au cours d'une opération ;

f. Il peut manquer des données, il peut y en avoir de surabondantes, voire de conflictuelles ;

g. Il peut y avoir plusieurs façons de s'y prendre, l'essentiel étant d'atteindre le but en un temps acceptable ; en corollaire, il y a aussi risque d'échec ;

h. S'il y a échec, on peut recommencer et, éventuellement, réussir ;

i. L'on recourt à des recettes plus qu'à des méthodes : l'algorithme cède

9. *Godel, Escher, Bach, les Brins d'une guirlande éternelle*, InterÉditions. D. Hofstadter a tenu à écrire en français l'avant-propos de son livre "... car, de toutes les langues cibles, il n'y en a qu'une que j'aime suffisamment pour vouloir, pouvoir, voire *devoir* en faire la langue de mon propre avant-propos".

la place à l'heuristique ; le "truc" est, dans l'échelle des valeurs, souvent au-dessus de la raison pure, l'heuristique n'est pas un algorithme au rabais ;

j. Il y a, pour une opération donnée, apprentissage au fur et à mesure des essais d'exécution successifs, réussis ou ratés ;

k. Le système peut être capable d'évaluer l'efficacité de son action et de changer de comportement ;

l. Il peut y avoir collaboration entre l'intelligence propre du système et celle de l'opérateur humain (dimension sociale de l'intelligence) ; ce point essentiel constitue une découverte assez récente ;

m. Lorsqu'il s'agit de robotique, les robots ont des organes sensoriels pour appréhender l'environnement et des organes moteurs (effecteurs) matérialisant leur autonomie ; leur "intelligence" réside habituellement dans un ensemble séparé de traitement : ils n'ont pas, comme les hommes, un ordinateur individuel portatif...

n. L'idéal, pour la pratique de l'I.A., est de disposer d'un ordinateur et d'un langage (LISP, PROLOG,...) permettant :

La représentation interne de connaissances et de règles de natures très diverses ;

L'introduction de ces connaissances en vrac, indépendamment de leur utilisation future (programmation déclarative) et en langage aussi peu informatique que possible ;

L'adjonction ou la modification de connaissances sans refonte totale du programme ;

Un dialogue agréable entre le système et l'opérateur ;

La manipulation tous azimuts des connaissances et des données particulières du problème à résoudre : déductions, inductions, en un mot inférences.

La liste n'est pas close, mais suffira à notre bonheur de néophytes. Notons, comme pour la liste d'applications des pages 26 et 27, l'impression curieuse ressentie à sa lecture : son caractère hétéroclite, à la fois incomplet et surabondant ; il y a de la mode et de l'éphémère là-dedans.

Nous essaierons d'en trouver les raisons en déroulant les exemples : les robots nous diront peut-être pourquoi la lueur d'intelligence du vendredi prend souvent, le dimanche, une bien terne apparence...

Pour terminer ce chapitre, nous donnerons la définition de l'I.A. proposée par Alain Bonnet dans son livre déjà cité :

"C'est la discipline visant à comprendre la nature de l'intelligence en construisant des programmes d'ordinateur imitant l'intelligence humaine."

Cette définition est belle, car désintéressée. Dans la traduction commerciale de cette I.A. idéale, il est nécessaire d'inverser les termes, c'est-à-dire les buts et les moyens : construire et vendre des programmes d'ordinateur imitant l'intelligence, en essayant à cette fin de comprendre vaille que vaille la nature de l'intelligence humaine.

Mais cela n'est pas une critique, car les applications pratiques ont toujours été les meilleurs témoins et les moteurs les plus puissants des grandes aventures de l'esprit ; la pensée est un besoin physique comme un autre.

IV

QUE LA LUMIÈRE SOIT

''Ressemblons-leur : c'est le moyen d'avoir la paix.''
J. GREEN.

Ce chapitre est consacré à l'émergence progressive du comportement "intelligent" d'un robot mobile, à partir de la passivité inerte d'un projectile. La métamorphose passera par une belle chrysalide aux talents intermédiaires : un missile autoguidé.

Cette évolution cérébrale, au cours de laquelle l'objet mobile acquerra un standing ascendant et méritera des noms de plus en plus flatteurs, s'opérera en sept étapes, à travers sept exemples simples permettant d'en saisir avec précision les stades.

Chaque exemple, repéré de E1 à E7, sera d'abord exposé en langage naturel, puis interprété en BASIC. Ils sont facilement transposables sur d'autres micro-ordinateurs. Les conditions de cette interprétation seront chaque fois soigneusement précisées — notamment les moyens sensoriels dont le robot est censé disposer — afin qu'il y ait véritablement un *isomorphisme* [1] aussi serré que possible entre le scénario réel et le système correspondant de simulation sur l'ordinateur, avec ses instructions et son écran.

Il est en effet indispensable, pour tirer des conclusions valables du spectacle de chaque aventure, de savoir exactement ce qu'on y a fait entrer. Nous verrons que cela est souvent facile, mais qu'il y a néanmoins des cas où le transfert réalité-ordinateur est plus délicat : le risque principal d'erreur ne consiste pas à *oublier* d'insuffler au robot les facultés qu'on entend lui donner, mais au contraire à lui en donner trop, sans se rendre compte que certaines opérations de l'ordinateur, la reconnaissance de la direction géographique du maître par exemple, impliquent l'existence chez le robot, non seulement du sens de la vue (détecter la présence d'un objet) mais aussi du sens des angles et de leur estimation...

Dans certains exemples ultérieurs (E13, E27, E28, E30), cela ira plus loin : il faudra que ce sens de la vue comprenne un goniomètre comme précédemment et un télémètre pour apprécier les distances.

Bien entendu, cet isomorphisme se traduit dans l'ordinateur par des calculs à partir des coordonnées des divers éléments concernés se traduisant eux-mêmes par des positions affichées sur l'écran.

Mais tout cela s'éclairera facilement sur les exemples, ainsi que bien d'autres opérations de représentation de l'univers de nos robots ; c'est ainsi que nous les doterons de la vue, du toucher, et même de l'ouïe... et de l'odorat (E26).

Il va de soi que sur de vrais robots, il est moins facile d'installer des

1. *Isomorphisme* : correspondance entre les éléments de deux structures ; chaque élément de l'une donne lieu à une *interprétation* de chaque élément correspondant de l'autre. Un isomorphisme bien choisi permet ainsi de représenter une réalité (de vrais robots) par un système de symboles (l'ordinateur).

facultés sensorielles de ce genre. L'odorat en particulier n'a pas encore trouvé son palpeur valable — l'équivalent de la caméra pour la vue ou du microphone pour l'ouïe —, s'agissant de déceler d'infimes traces en suspension dans l'air ambiant.

Nos simulations sont heureusement dispensées de ces difficultés de première grandeur.

ÉTAPE E1

L'INERTIE AVEUGLE

Présentation

Un projectile est tiré au hasard dans une direction quelconque, sur un terrain dégagé, dans l'espoir qu'il atteigne une cible fixe dont la position est également déterminée au hasard avant chaque tir.

Pour rester dans les limites de l'écran, le projectile rebondit indéfiniment sur les bords comme une boule de billard. Il est donc doté d'un sens du toucher qui lui sert également lors du contact avec la cible. Il est évidemment nécessaire de simuler ce sens dans le programme informatique : c'est le rôle des deux sous-programmes "effets de bords" et "cible atteinte".

Ce premier scénario a deux objectifs : indiquer la structure de principe de tous les programmes du livre et, surtout, tirer les conclusions du comportement de ce projectile primitif.

La structure d'un programme de simulation de mouvement pas à pas est simple. Elle est décrite avec plus de détails dans le livre *Cinématique-Dynamique* déjà cité au Chapitre II.

En résumé, elle consiste essentiellement :

- *en une première zone concernant les données de départ*

  Fixer et afficher sur l'écran le point de départ du mobile.

  Fixer éventuellement la vitesse et la direction du lancement (si le mobile est lancé comme c'est le cas).

  Fixer les données du terrain : dans le présent exemple, s'agissant d'un terrain nu, seule la cible est à prendre en considération.

- *en une deuxième zone de génération des mouvements*

  Constituer *la boucle des mouvements élémentaires du mobile*, partie clé du programme. L'instruction principale de cette boucle est celle qui donne la nouvelle position à partir de la position précédente ; on ajoute simplement aux coordonnées précédentes les composantes de la vitesse. Celle-ci est bien entendu préalablement calculée compte tenu de tous les éléments du scénario (position de la cible, obstacles, mode de "raisonnement" du mobile, etc.). Dans le cas présent, comme il s'agit d'un projectile inerte, il conserve tout bonnement sa vitesse de lancement.

La boucle des mouvements renvoie par ailleurs aux divers sous-programmes susceptibles de jouer un rôle :

Effets correctifs sur la position ou la vitesse du mobile, par exemple les effets de bords si le projectile dépasse les limites de l'écran ; on l'y ramène alors et l'on change le signe de la composante de vitesse responsable du débordement, simulant ainsi la loi physique d'un rebondissement. (Dans les programmes suivants, il y aura ainsi également un effet correctif chaque fois que le mobile touchera un obstacle.)

Effets de sortie, par exemple arrêt du parcours lorsque la cible est atteinte, et renvoi à un nouveau tir.

La boucle des mouvements comporte enfin l'affichage sur l'écran de la nouvelle position du mobile et, si désiré, l'effacement de la précédente ou l'affichage de sa trace atténuée (teinte pastel) pour visualiser agréablement la trajectoire. Cette boucle FOR NEXT doit comporter un nombre de mouvements potentiels suffisant pour permettre le déroulement complet du scénario : 500 dans le présent exemple.

- *en une troisième zone de sous-programmes*

  Élaborer les sous-programmes nécessaires. Dans le présent exemple, il y en a deux, l'un pour les effets de bords, l'autre pour vérifier à chaque pas si la cible est atteinte ; ces deux sous-programmes utilisent le sens du toucher dont notre missile a été doté.

Pour chaque exemple, et en particulier celui-ci, on trouvera le programme démonstratif ainsi que ses commentaires techniques éventuels en fin de l'exposé.

Noter que les remarques incluses dans les programmes proprement dits (REM) permettent d'en saisir déjà l'essentiel.

Quelques effets sonores ou visuels accessoires visent à donner du sel à l'action ou ont un rôle didactique : visualisation des piliers du labyrinthe touchés, points d'interrogation sur les positions susceptibles d'être choisies, etc.

Venons-en maintenant, après ces quelques considérations techniques, aux enseignements de ce premier exemple de comportement d'un objet mobile. Il n'est pas difficile d'en constater l'ineptie et l'inefficacité. Bien que les multiples rebondissements aux limites de l'écran entretiennent les chances de ratissage, la réussite de la mission s'avère incertaine.

L'on voit, d'après les dimensions de l'écran, qu'il y a un millier de positions possibles pour la cible. Le programme permet au projectile, comme nous l'avons dit, d'accomplir 500 pas. Dans ces conditions, la

probabilité de réussite est de l'ordre de 40 pour 100. Les lecteurs avertis en calcul de probabilités pourront l'évaluer exactement [2].

Plus prosaïquement, le programme proposé permet de *mesurer* cette probabilité : c'est le quotient RGR/RG (RGR, nombre de parcours réussis ; RG, nombre total des parcours effectués).

Ce résultat peut paraître curieux, un taux de réussite aussi élevé avec un comportement aussi stupide !... Et 500 pas, c'est vite fait. Bien entendu, cela est lié aux ratissages multiples dus aux rebondissements. Dans la réalité d'un terrain illimité, il y aurait une probabilité infinitésimale d'atteindre le but et une quasi-certitude que le projectile se perde dans l'infinité des horizons possibles.

Mais peut-être est-il moins évident de remarquer que, dans le cas d'un espace limité tel que l'écran, il n'est pas plus idiot de partir au hasard que de mal viser : la probabilité de réussite est la même. Seule une visée rigoureuse assure la victoire (étape E3 ci-après). Pour le vérifier, il est facile de modifier le programme en laissant l'opérateur estimer le cap d'un coup d'œil puis l'introduire au clavier à l'aide d'une instruction INPUT ou INKEY$.

On constate ainsi que l'essentiel, dans une telle aventure, est moins *d'y aller* que *d'aller* : un bon moyen d'améliorer une mauvaise position consiste à en changer jusqu'à ce qu'on en trouve éventuellement une bonne.

L'intelligence humaine, donc l'I.A., utilise couramment des lapalissades de ce genre auxquelles on a donné un nom distingué que nous découvrirons à l'étape E4.

Programme

```
100 'ETAPE E1:PROJECTILE INERTE,CIBLE FI
XE,LANCEMENT AU HASARD
110 INK 0,22 :INK 1,1 :INK 2,6 :CLS :CLE
AR :RG=0 :RGR=0 'NB DE PARCOURS EFFECTUE
S RG ET NB DE PARCOURS REUSSIS RGR
120 'NOMBRES ALEATOIRES VARIES
130 RANDOMIZE TIME
140 CLS :H=INT(2*RND) :IF H=1 THEN PXC=I
NT(10*RND+7) :PYC=5 ELSE PXC=7 :PYC=INT(
10*RND+7) 'TIRAGE POSITION CIBLE
150 PXPRO=28 :PYPRO=23 :PPXPRO=PXPRO :PP
YPRO=PYPRO 'POS DEP PROJECTILE ET MISE E
N RESERVE POUR EFFACMT
```

2. Attention ! La probabilité de réussite n'est pas de 50 pour 100 au bout de 500 pas, même si elle est de 0,1 pour 100 (en moyenne) au bout de chaque pas.

```
160 PEN 1 :LOCATE PXPRO,PYPRO :PRINT"*"
:PEN 2 :LOCATE PXC,PYC:PRINT"o" 'AFF PRO
JECTILE ET CIBLE
170 CAPPRO=INT(360*RND) :VXPRO=SIN(CAPPR
O/57.3) :VYPRO=-COS(CAPPRO/57.3) 'TIRAGE
 CAP DEPART DE 0 A 359 DEG ET CALC COMP
VITESSE PROJECTILE
180 SOUND 1,200,80,7 :FOR AT=1 TO 2000 :
NEXT 'TOP DEPART
1000 REM BOUCLE DES MOUVEMENTS
1010 FOR M=1 TO 500
1020 PXPRO=PXPRO+VXPRO :PYPRO=PYPRO+VYPR
O 'CALC NOUV POSITION
1030 GOSUB 3000 'EFFETS DE BORDS
1040 PEN 1 :IF CINT(PXPRO)<>40 OR CINT(P
YPRO)<>25 THEN LOCATE PPXPRO,PPYPRO :PRI
NT" " :LOCATE PXPRO,PYPRO :PRINT"*" :SOU
ND 1,30,1,7 :PPXPRO=PXPRO :PPYPRO=PYPRO
ELSE SOUND 1,53,10,7 :PXPRO=39 :VXPRO=-A
BS(VXPRO) :VYPRO=-ABS(VYPRO) :GOTO 1040'
AFF
1050 GOSUB 4000'CIBLE ATTEINTE
1060 NEXT M
1070 REM PARCOURS RATE
1080 SOUND 1,800,50,7 :FOR AT=1 TO 1000:
NEXT AT :LOCATE PPXPRO,PPYPRO :PRINT" "
:RG=RG+1 :GOTO 150
1090 END
2000 REM ZONE DES SOUS-PROGRAMMES
3000 REM SSPROG EFFETS DE BORDS
3010 IF CINT(PXPRO)>0 THEN 3020 ELSE SOU
ND 1,53,10,7 :PXPRO=1 :VXPRO=-VXPRO
3020 IF CINT(PXPRO)<41 THEN 3030 ELSE SO
UND 1,53,10,7 :PXPRO=40 :VXPRO=-VXPRO
3030 IF CINT(PYPRO)>0 THEN 3040 ELSE SOU
ND 1,53,10,7 :PYPRO=1 :VYPRO=-VYPRO :RET
URN
3040 IF CINT(PYPRO)<26 THEN RETURN ELSE
SOUND 1,53,10,7 :PYPRO=25 :VYPRO=-VYPRO
:RETURN
4000 REM SSPROG CIBLE ATTEINTE
4010 IF CINT(PXPRO)<>PXC THEN RETURN ELS
E IF CINT(PYPRO)<>PYC THEN RETURN
4020 FOR FOIS=1 TO 4 :SOUND 1,100,10,7 :
FOR AT=1 TO 100:NEXT :NEXT :FOR AT=1 TO
500:NEXT :RG=RG+1 :RGR=RGR+1 :GOTO 140
```

Commentaires techniques

- Tirage au hasard des positions et des vitesses : l'appel au générateur de nombres aléatoires se fait suivant des modalités qui diffèrent d'un

ordinateur à l'autre. Sur l'Amstrad, cela s'effectue grâce à l'instruction RANDOMIZE.

- Identification des variables : on les désigne par des lettres ou des groupes de lettres évocateurs et rationnels pour faciliter la programmation, par exemple PX et PY pour la position, VX et VY pour la vitesse, PPX et PPY pour la position précédente, etc.

ÉTAPE E2

DES OBSTACLES INUTILES

Présentation

Il s'agit du même projectile inerte et aveugle. Mais on a introduit sur le terrain des obstacles, placés au hasard, sur lesquels le projectile rebondit dans une direction quelconque. Il est donc doté, comme en E1, du sens du toucher.

On peut constater que, si la présence des obstacles n'améliore pas les performances du projectile, elle ne les détériore pas non plus complètement.

Le hasard, total au départ, est assez peu sensible à de nouveaux facteurs de hasard.

Le néant a ceci de fort qu'il ne saurait être diminué.

Constatons donc, là encore, que la cible n'est pas totalement à l'abri, du seul fait que le projectile, aussi stupide soit-il, rôde dans les parages et ne se laisse arrêter ni par les bords, ni par les obstacles.

C'est la seule miette d'intelligence que l'on puisse lui reconnaître, mais c'est une grosse miette...

Programme

```
100 'ETAPE E2:PROJECTILE INERTE,CIBLE FI
XE,LANCEMENT AU HASARD,OBSTACLES,REBONDS
 AU HASARD
110 INK 0,22:INK 1,1:INK 2,6:CLS:CLEAR

120 DIM PXPI(30),PYPI(30) 'TABLEAUX OBST
ACLES
130 REM SERIES VARIABLES DE NB ALEATOIRE
S
140 RANDOMIZE TIME
150 CLS:PXC=INT(20*RND+10):PYC=INT(8*RND
+10):PXPRO=PXC+INT(11*RND-5):PYPRO=PYC+I
NT(11*RND-5):IF PYPRO=PYC THEN PYPRO=PYC
+1'POS DEP PROJECTILE
160 PPXPRO=PXPRO:PPYPRO=PYPRO'MISE EN RE
S POUR EFFT
170 CLS:PEN 1:LOCATE PXPRO,PYPRO:PRINT "
*":PEN 2:LOCATE PXC,PYC:PRINT "o"'AFF PR
OJ ET CIBLE
```

```
180 FOR N=1 TO 30 :PXPI(N)=PXC+INT(11*RN
D-5) :PYPI(N)=PYC+INT(11*RND-5) :IF PXPI
(N)=PXPRO THEN IF PYPI(N)=PYPRO THEN PXP
I(N)=0 :GOTO 210
190 IF PXPI(N)=PXC THEN IF PYPI(N)=PYC T
HEN PXPI(N)=0 :GOTO 210
200 PEN 1 :LOCATE PXPI(N),PYPI(N) :PRINT
 "=" 'TIRAGE ET AFF 30 OBSTACLES
210 NEXT N
220 CAPPRO=INT(360*RND):VXPRO=SIN(CAPPRO
/57.3):VYPRO=-COS(CAPPRO/57.3)'TIRAGE CA
P DEPART DE 0 A 359 DEG ET CALC COMP VIT
ESSE PROJECTILE
230 SOUND 1,200,80,7:FOR AT=1 TO 2000:NE
XT'TOP DEPART
1000 REM BOUCLE DES MOUVEMENTS
1010 FOR M=1 TO 500
1020 PXPRO=PPXPRO+VXPRO:PYPRO=PPYPRO+VYP
RO'CALC NOUV POSITION
1030 GOSUB 3000'EFFETS DE BORDS
1040 GOSUB 4000'OBSTACLE
1050 IF OBS=1 THEN 1020
1060 PEN 1:IF CINT(PXPRO)<>40 OR CINT(PY
PRO)<>25 THEN LOCATE PPXPRO,PPYPRO:PRINT
" ":LOCATE PXPRO,PYPRO:PRINT"*":SOUND 1,
30,1,7:PPXPRO=PXPRO:PPYPRO=PYPRO ELSE SO
UND 1,53,10,7:PXPRO=39:VXPRO=-ABS(VXPRO)
:VYPRO=-ABS(VYPRO):GOTO 1060'AFF ET MISE
 EN RES
1070 GOSUB 5000'CIBLE ATTEINTE
1080 NEXT M
1090 END
2000 REM ZONE DES SOUS-PROGRAMMES
3000 REM SSPROG EFFETS DE BORDS
3010 IF CINT(PXPRO)>0 THEN 3020 ELSE SOU
ND 1,53,10,7:PXPRO=1:VXPRO=-VXPRO
3020 IF CINT(PXPRO)<41 THEN 3030 ELSE SO
UND 1,53,10,7:PXPRO=40:VXPRO=-VXPRO
3030 IF CINT(PYPRO)>0 THEN 3040 ELSE SOU
ND 1,53,10,7:PYPRO=1:VYPRO=-VYPRO:RETURN
3040 IF CINT(PYPRO)<26 THEN RETURN ELSE
SOUND 1,53,10,7:PYPRO=25:VYPRO=-VYPRO:RE
TURN
4000 REM SSPROG OBSTACLE
4010 OBS=0 'ANNUL OBST PRECDT
4020 IF CINT(PXPRO)<PXC-5 THEN RETURN EL
SE IF CINT(PXPRO)> PXC+5 THEN RETURN  EL
SE IF CINT(PYPRO)<PYC-5 THEN RETURN ELSE
 IF CINT(PYPRO)>PYC+5 THEN RETURN
4030 FOR N=1 TO 30 :IF CINT(PXPRO)<>PXPI
(N) THEN 4060 ELSE IF CINT(PYPRO)<>PYPI(
N) THEN 4060 ELSE SOUND 1,80,10,7 :PEN 2
 :LOCATE PXPRO,PYPRO:PRINT "." :FOR AT=1
 TO 90:NEXT :PEN 1 :LOCATE PXPRO,PYPRO:P
```

```
RINT "="
4040 S1=INT(3*RND-1) :IF S1<>0 THEN S2=I
NT(3*RND-1) ELSE S2=INT(2*RND)-0.5 'TIRA
GE COMPOSANTES NOUV VITESSE
4050 VXPRO=SGN(S1) :VYPRO=SGN(S2) :OBS=1
 :N=30
4060 NEXT N
4070 RETURN
5000 REM SSPROG CIBLE ATTEINTE
5010 IF CINT(PXPRO)<>PXC THEN RETURN ELS
E IF CINT(PYPRO)<>PYC THEN RETURN
5020 FOR FOIS=1 TO 4:SOUND 1,100,10,7:FO
R AT=1 TO 100:NEXT AT:NEXT FOIS:FOR AT=1
 TO 500:NEXT AT:GOTO 140
```

Commentaires techniques

- Les 30 obstacles sont introduits au moyen de *variables indicées* : les coordonnées PXPI et PYPI des piliers, tirées au hasard, sont indicées de 0 à 29, ce qui permet de les mémoriser individuellement et de savoir par la suite si le projectile touche l'un deux.

  C'est à cette fin que le sous-programme obstacle passe systématiquement en revue les 30 positions de piliers pour voir si l'une d'elles coïncide avec celle du projectile. Dès qu'il a détecté un pilier, les recherches sont interrompues.

  Cette façon de procéder est facile à saisir, mais elle est longue à l'exécution comme on peut le constater, car il faut faire 30 boucles de test après chaque pas du projectile. Nous utiliserons à partir de E9 une autre méthode plus puissante faisant appel à la notion de prédicat, mieux en harmonie avec les méthodes générales de l'I.A., pour déterminer instantanément s'il y a un obstacle ou non à tel ou tel endroit.

- Noter, dans la boucle des mouvements, que PX et PY sont calculés à partir de la position précédente mémorisée PPX et PPY, et non à partir de PX et PY ; cela est indispensable car, en cas de contact avec un pilier, une autre position doit être trouvée et recalculée à partir de la position de départ.

- On fait intervenir dans le sous-programme obstacle la variable OBS, qui prend la valeur 1 s'il y a un obstacle et 0 si la place est libre. Il s'agit là d'une *proposition logique* vraie ou fausse, courante en programmation. Elle est aussitôt utilisée comme une sorte de drapeau de signalement dans la boucle des mouvements, dès le retour du sous-programme : si OBS = 0, l'on passe ; si OBS = 1, l'on recalcule la nouvelle position visée à l'aide de la nouvelle vitesse fournie par le sous-

programme et l'on recommence le cérémonial. Il est nécessaire de ramener OBS à Ø à chaque entrée dans le sous-programme pour éviter les obstacles imaginaires.

- Le programme ne renvoie au sous-programme que si le projectile se trouve effectivement dans la zone où tous les obstacles ont été volontairement concentrés, cela accélère l'exécution.

ÉTAPE E3

UN PREMIER CLIN D'ŒIL

Présentation

Une première lueur mentale jaillit, unique et définitive à l'instant du lancement, puis c'est à nouveau le naufrage dans la passivité pour le reste du parcours. En effet, on a cette fois affaire à un projectile *autolancé* sur cible fixe et terrain dégagé.

La faculté d'autolancement consiste pour le projectile à percevoir avant le départ la cible et sa direction géographique, puis à foncer dans cette direction et à s'y tenir sans autre vérification ultérieure. L'impression quelquefois zigzagante de la course est due à la faible définition des positions d'écran (arrondis d'affichage).

La simulation sur l'ordinateur de ce sens goniométrique de la vue se réalise par le calcul trigonométrique (tangente, puis cosinus et sinus) des composantes de la vitesse du projectile, calcul basé sur les positions respectives de la cible et du projectile au moment du lancement.

Cette évaluation n'est donc faite qu'une seule fois et l'on peut imaginer que, dans la réalité, l'''œil'' du projectile resterait sur la base de lancement pour les prochains tirs, économie oblige.

On peut admirer à l'exécution la fiabilité, c'est-à-dire l'infaillibilité absolue du système, grâce à la précision des calculs (6 décimales). Cela suppose bien sûr pour notre projectile un sens de la vue et des angles très performant, dont la réalisation concrète serait probablement assez difficile.

Notons que les calculs conservent comme précédemment leur formalisme mathématique, y compris une précaution contre la division par zéro, dont chacun de nous se soucie pourtant comme d'une guigne dans la vie courante [3]...

3. La division par zéro correspond pourtant dans la vie courante à une savoureuse absurdité : partager la succession d'un défunt qui n'aurait aucun héritier, ce qui est d'autant plus dommage que la part de chacun est alors infinie ; l'État, avide, se précipite sur le pactole et celui-ci reprend aussitôt une dimension terrestre qui n'a plus rien de miraculeux.

Programme

```
100 'ETAPE E3:PROJECTILE INERTE,CIBLE FI
XE,TERRAIN NU,AUTO-LANCEMENT
110 INK 0,22:INK 1,1:INK 2,6:CLS:CLEAR
120 'NOMBRES ALEATOIRES VARIES
130 RANDOMIZE TIME
140 CLS:H=INT(2*RND):IF H=1 THEN PXC=INT
(10*RND+7):PYC=5 ELSE PXC=7:PYC=INT(10*R
ND+7)'TIRAGE POSITION CIBLE
150 PXPRO=28:PYPRO=23:PPXPRO=PXPRO:PPYPR
O=PYPRO'POS DEP PROJECTILE ET MISE EN RE
SERVE POUR EFFT
160 PEN 1:LOCATE PXPRO,PYPRO:PRINT"*":PE
N 2:LOCATE PXC,PYC:PRINT"o"'AFF PROJECTI
LE ET CIBLE
170 REM CALC INCLINAISON IN DE LA DIR DE
 LA VITESSE DE DEPART DU PROJECTILE
180 NUME=ABS(PYPRO-PYC) :DENO=ABS(PXPRO-
PXC) :IF DENO<0.1 THEN DENO=0.1
190 IN=NUME/DENO
200 REM CALC COMPOSANTES VITESSE
210 AN=ATN(IN) :VXPRO=COS(AN)*SGN(PXC-PX
PRO) :VYPRO=SIN(AN)*SGN(PYC-PYPRO)
220 SOUND 1,200,80,7:FOR AT=1 TO 2000:NE
XT'TOP DEPART
1000 REM BOUCLE DES MOUVEMENTS
1010 FOR M=1 TO 500
1020 PXPRO=PXPRO+VXPRO:PYPRO=PYPRO+VYPRO
'CALC NOUV POSITION
1030 PEN 1:LOCATE PPXPRO,PPYPRO:PRINT" "
:LOCATE PXPRO,PYPRO:PRINT"*":SOUND 1,30,
1,7:PPXPRO=PXPRO:PPYPRO=PYPRO'AFF ET MIS
E EN RES
1040 GOSUB 3000'CIBLE ATTEINTE
1050 NEXT M
1060 END
2000 REM ZONE DES SOUS-PROGRAMMES
3000 REM SSPROG CIBLE ATTEINTE
3010 IF CINT(PXPRO)<>PXC THEN RETURN ELS
E IF CINT(PYPRO)<>PYC THEN RETURN
3020 FOR FOIS=1 TO 4:SOUND 1,100,10,7:FO
R AT=1 TO 100:NEXT:NEXT:FOR AT=1 TO 500:
NEXT:GOTO 140
```

Commentaires techniques

Dans le sous-programme cible atteinte, il faut prendre la valeur *arrondie* des coordonnées du projectile pour en tester l'identité avec les coordonnées de la cible, car les composantes de la vitesse sont, comme nous l'avons vu, calculées avec plusieurs décimales.

ÉTAPE E4

LES HEURISTIQUES

Présentation

Il s'agit d'un *missile autoguidé* sur cible fixe, doté comme le projectile de E3 d'un sens de la vue directionnel. Mais son comportement mental est cette fois radicalement différent.

Peu doué pour les calculs, il préfère avancer d'une manière plus approximative vers la cible mais en y regardant tout au long du parcours, faisant en quelque sorte le point après chacun de ses pas. Théoriquement, nous avons vu que cela n'est pas nécessaire si l'on vise parfaitement la cible au départ, mais cela ne coûte rien et l'on ne sait jamais...

Ici intervient la notion d'*heuristique*, la pensée pratique qui prime sur les considérations de raison pure. En l'occurrence, on va faire le point *surabondamment* et s'appuyer sur cette redondance pour assurer un succès que le caractère sommaire de la démarche ci-après risquerait de compromettre.

Cette démarche, qui s'apparente au système D, dédaigne le concept pourtant imparable consistant à se diriger droit sur la cible car il prend en compte les éléments suivants :

- il est fatiguant de mesurer exactement la direction et de tenir un même cap tout au long du parcours ;
- de toute façon, ce n'est pas possible de manière rigoureuse puisque, à chaque pas, le missile n'a le droit de se placer que sur l'une des huit positions (cases de l'écran caractérisées par un numéro de ligne et un numéro de colonne) voisines de celle qu'il occupe ; il ne peut donc faire que des pas horizontaux, verticaux ou en diagonale.

Comme le montre la figure ci-après, il peut, au mieux, effectuer le parcours A du programme précédent, E3, qui n'est que pseudo-rectiligne car il serpente autour de la droite idéale du fait des arrondis d'affichage. Il est inutile dans ces conditions de passer du temps à calculer la direction à un millième de degré près, ce qui retarde l'action et la victoire.

Enfin, argument décisif et inattendu, le nombre de pas à effectuer avec le parcours pseudo-rectiligne A est exactement le même que si l'on choisit plus simplement un parcours brisé (parcours B ou parcours C) facile-

ment déterminable sans calcul compliqué comme nous allons le voir [4].

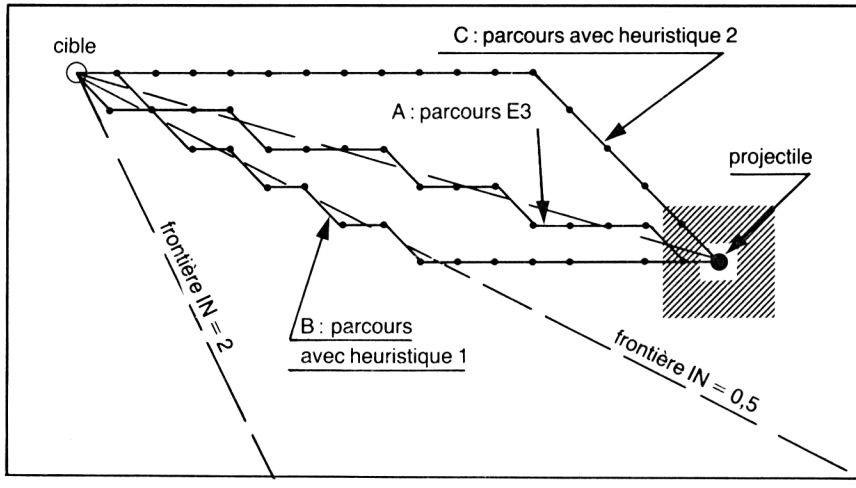
Une heuristique est ainsi une manière empirique de penser pour résoudre un problème, pour faire face à une situation, par opposition à l'*algorithme* de l'étape E3 qui menait mathématiquement à la solution [5].

L'heuristique peut avoir l'inconvénient de ne pas toujours aboutir (ce n'est pas le cas ici) et de ne pas toujours satisfaire un esprit rigoureux, mais là n'est pas le but. Cela est compensé par plusieurs avantages.

Le premier est celui, en général, de la simplicité, le seul principe directeur, souvent insconscient, étant celui de la moindre fatigue intellectuelle possible, même si dans certains cas il y a en rançon un travail matériel un peu plus long ou un peu plus pénible. C'est en quelque sorte la somme de ces deux catégories d'efforts qu'une bonne heuristique cherche à minimiser.

Le second avantage est un temps de réflexion moins long entre deux phases d'action successives. Par exemple, l'heuristique 1 du parcours B permet d'atteindre la cible plus vite que les calculs du parcours A de E3, et l'heuristique 2 du parcours C, particulièrement simple dans sa conception, permet d'y arriver encore plus rapidement. Rappelons que, dans les trois cas, le nombre de pas est équivalent, c'est la cadence qui est différente.

Nous présenterons vers la fin du livre, en E33, un *banc de comparaison* des heuristiques d'une épreuve donnée, tenant compte de cette considération essentielle.



4. Une formulation générale de cette constatation et de ses conséquences est proposée en Annexe III, après avoir pris un éclairage plus précis en E13.

5. *Cf.* P. Beaufils et W. Luther, *Algorithmes*, Éditions Sybex.

Voyons maintenant de façon précise ce que sont les deux heuristiques imaginées pour notre missile, étant bien entendu qu'il en existe beaucoup d'autres possibles, que le lecteur pourra inventer. C'est en ce domaine des heuristiques que l'imagination révèle la place qu'elle tient dans l'intelligence.

Réglons d'abord une question de vocabulaire utile pour la suite. Il s'agit pour le projectile de se déplacer en *terrain libre* de manière à atteindre la cible dans les meilleures conditions. Nous conviendrons dans ce cas que nous avons affaire à une *heuristique de navigation*, par opposition aux *heuristiques de dégagement* que les robots utiliseront plus loin pour contourner les obstacles placés sur leur chemin.

Les heuristiques de navigation de ce livre seront repérées par un chiffre de 1 à 6 et les heuristiques de dégagement par une lettre de A à I. Toutes ces heuristiques figurent dans un tableau récapitulatif en fin d'ouvrage (Annexe I).

Cela dit, la première heuristique repérée 1 consiste, en chaque point du parcours, à apprécier l'inclinaison de la direction missile-cible par rapport à l'horizontale (direction ouest-est de l'écran), ce qui est simulé sur l'ordinateur par le calcul du rapport IN = distance verticale missile-cible/distance horizontale missile-cible.

Par exemple, sur la figure, ce rapport vaut 5/17 = 0,29.

Suivant la valeur du rapport IN, la décision du missile est la suivante :

- si IN est inférieur à 0,5 (c'est-à-dire si le missile se trouve à droite de la frontière IN = 0,5 tracée sur la figure), le missile fait un pas horizontalement ;
- si IN est supérieur à 2 (c'est-à-dire si le missile se trouve au-dessous de la frontière IN = 2), il fait un pas verticalement ;
- si IN est compris dans la fourchette de 0,5 à 2 (c'est-à-dire si le missile se trouve dans le secteur intermédiaire), il fait un pas en diagonale.

Le programme informatique correspondant, aussi bien que les opérations mentales humaines qu'il faudrait effectuer, sont d'exécution beaucoup plus rapide que ceux de l'étape E3 ; un coup d'œil, une estimation de l'angle (une légère erreur n'a aucun effet dommageable sur le résultat final !), et une décision simple : vers la gauche, vers le haut, en biais.

Quant à l'heuristique de navigation 2, c'est le modèle idéal en matière de simplicité et de rapidité. Il n'y a rien à calculer. Si le missile et la cible sont sur une même ligne, un pas horizontal ; si le missile et la cible sont sur une même colonne, un pas vertical ; dans tous les autres cas, un pas en biais.

Moins formelles que les algorithmes, les heuristiques possèdent parfois l'étrange beauté de l'indéfinissable à côté de l'ascétique splendeur de certains cheminements mathématiques.

Les trajets B et C correspondant aux heuristiques de navigation 1 et 2 sont représentés sur la figure, ainsi que le trajet A de l'étape E3. Ils sont, comme on le voit, très différents.

Le trajet B est d'abord horizontal jusqu'à la frontière IN = 0,5 puis il suit cette frontière en serpentant au plus près.

Le trajet C est apparemment moins direct mais comporte cependant le même nombre de pas que A et B ; il présente en outre une allure *plus décidée* qui apparaît nettement à l'exécution en raison de la cadence élevée des mouvements. Il suffit en effet au missile, dans la "réalité", de voir à chaque pas si sa visée de la cible est parallèle ou non à l'un des bords du terrain, en imaginant par exemple deux mires lointaines situées sur le prolongement de chacun des bords nord et ouest.

C'est ce que fait le programme présenté ci-après, qui exécute à titre de comparaison successivement la même mission (mêmes positions de départ) avec les heuristiques 1 puis 2.

On dit en Sologne que le poireau est l'asperge du pauvre. Outre que ce dicton tendancieux est faux, car le poireau a ses propres vertus, il n'est absolument pas transposable à l'heuristique vis-à-vis de l'algorithme. L'heuristique n'est nullement un algorithme au rabais, à moins de considérer l'homme comme inférieur à la machine.

On pourrait à la rigueur prétendre que l'heuristique est un algorithme de paresseux futé, le second qualificatif étant largement suffisant au pardon du premier, et le tout constituant le glorieux label que beaucoup revendiquent.

À ce stade E4, il ne s'agit donc plus tout à fait d'un projectile, mais sûrement pas encore d'un robot car les critères énumérés au Chapitre III sont encore loin d'être significativement réunis. Seuls y sont présents les critères *c*, *g*, *i* et *m*, les critères *i* et *m* étant les plus caractéristiques de l'I.A.

Programme

```
100 'ETAPE E4:MISSILE AUTO-GUIDE,CIBLE F
IXE,TERRAIN NU,1ER PARCOURS AVEC HEURIST
IQUE DE NAVIGATION (1),PUIS 2EME PARCOUR
S AVEC HEURISTIQUE DE NAVIGATION (2)
110 INK 0,22:INK 1,1:INK 2,6:CLS:CLEAR
120 SYMBOL AFTER 32:SYMBOL 33,&F0,&E0,&F
0,&B8,&1C,&F,&6,&4
130 RG=0'INITIALISATION RANG RG DU PARCO
URS
```

```
140 'NOMBRES ALEATOIRES VARIES
150 RANDOMIZE TIME
160 CLS :PEN 1 :LOCATE 16,5 :PRINT "HEUR
ISTIQUE (1)"
170 H=INT(2*RND):IF H=1 THEN PXC=INT(20*
RND+8):PYC=6 ELSE PXC=8:PYC=INT(15*RND+8
)'TIRAGE POSITION CIBLE
180 PXMI=28:PYMI=23:PPXMI=PXMI:PPYMI=PYM
I'POS DEP MISSILE ET MISE EN RESERVE
190 LOCATE PXMI,PYMI:PRINT CHR$(33):PEN
2:LOCATE PXC,PYC:PRINT "o"'AFF MISSILE E
T CIBLE
200 SOUND 1,200,80,7:FOR AT=1 TO 2000:NE
XT'TOP DEPART
210 RG=RG+1
1000 REM BOUCLE DES MOUVEMENTS
1010 REM 1ER PARCOURS:HEUR NAV (1);2EME
PARCOURS:HEUR NAV (2)
1020 FOR M=1 TO 500
1030 IF RG=2 THEN 1060 ELSE IF PXMI<>PXC
 THEN NUME=ABS(PYMI-PYC) :DENO=ABS(PXMI-
PXC) :IN=NUME/DENO ELSE 1070 'CALC INCLI
NAISON IN DE LA DIR DE LA VITESSE
1040 IF IN<0.5 THEN VXMI=SGN(PXC-PXMI) :
VYMI=0 :GOTO 1080
1050 IF IN>2 THEN VXMI=0 :VYMI=SGN(PYC-P
YMI) :GOTO 1080
1060 VXMI=SGN(PXC-PXMI) :VYMI=SGN(PYC-PY
MI) :GOTO 1080
1070 VXMI=0 :VYMI=SGN(PYC-PYMI)
1080 PXMI=PXMI+VXMI:PYMI=PYMI+VYMI'CALC
NOUV POS
1090 PEN 1:LOCATE PPXMI,PPYMI:PRINT" ":L
OCATE PXMI,PYMI:PRINT CHR$(33):SOUND 1,3
0,1,7:PPXMI=PXMI:PPYMI=PYMI'AFF ET MISE
EN RES
1100 GOSUB 3000'CIBLE ATTEINTE
1110 FOR AT=1 TO 20:NEXT:NEXT M
1120 END
2000 REM ZONE DES SOUS-PROGRAMMES
3000 REM SSPROG CIBLE ATTEINTE
3010 IF PXMI<>PXC THEN RETURN ELSE IF PY
MI<>PYC THEN RETURN
3020 FOR FOIS=1 TO 4:SOUND 1,100,10,7:FO
R AT=1 TO 100:NEXT:NEXT:FOR AT=1 TO 500:
NEXT:IF RG=2 THEN 110
3030 LOCATE 29,5:PRINT"2":GOTO 180
```

Commentaires techniques

- Contrairement à E3 où le projectile calculait sa vitesse une fois pour toutes au départ, c'est-à-dire en amont de la boucle des mouvements, la vitesse est ici redéterminée à chaque pas, dans la boucle même.
- Les composantes VX et VY de la vitesse ne peuvent avoir que des valeurs très simples :

  VX = − 1 et VY = 0 pour les pas horizontaux ;
  VX = 0 et VY = − 1 pour les pas verticaux ;
  VX = − 1 et VY = − 1 pour les pas en diagonale.

- Pour chaque épreuve, le programme réalise deux parcours : le premier (RG = 1) avec l'heuristique de navigation 1, le second (RG = 2) avec l'heuristique 2. La différence de complexité des opérations est évidente.
- Dans le sous-programme cible atteinte, il n'est plus nécessaire d'arrondir la position du missile, puisque ses coordonnées sont toujours des nombres entiers.

ÉTAPE E5

UN ENVIRONNEMENT VARIABLE

Présentation

Il s'agit du même missile autoguidé que précédemment, mais le degré de progression par rapport à E4 est de taille : la cible est mobile, suivant un mouvement rectiligne uniforme de direction quelconque tirée au hasard avant le départ.

Cette fois, la nécessité d'un guidage permanent à chaque position du missile et de la cible ne se discute pas.

On est dans le cas d'un environnement *variable en cours d'exécution* de la mission (critère *e* du Chapitre III). Le monde du missile, bien que toujours réduit à deux éléments, les limites du terrain et la cible, devient changeant.

Aveugle et inerte en E1 et E2, ouvrant un œil en E3, économe de ses efforts intellectuels en E4, il va lui falloir maintenant regarder en permanence le seul objet de son univers, la cible mouvante.

Curieusement, bien qu'une complication importante vienne ainsi d'apparaître, le procédé mental utilisé en E4 reste parfaitement valable ; aucun organe sensoriel supplémentaire, aucune intelligence supérieure n'est requise pour surmonter cette nouvelle épreuve. Le missile pourra tout aussi bien utiliser l'heuristique de navigation 1 ou 2 de E4, voire l'algorithme de E3.

La seule condition d'un succès à tous coups est étrangère à l'I.A. : il faut que le missile vole au moins aussi vite que la cible. C'est le cas dans le programme présenté.

Si la cible va un peu plus vite (ce qui est facile à réaliser en multipliant les deux composantes de sa vitesse par 1,1 ou 1,2), le succès n'est pas garanti. Si la cible va nettement plus vite, elle échappe systématiquement à la poursuite.

En somme, la puissance physique est une composante de l'intelligence. Pour confirmer ce point de vue, nous verrons en E27 qu'un surcroît d'intelligence du missile peut, en terrain fermé, surmonter ce handicap : c'est alors l'intelligence qui devient une composante de la puissance physique, ce qui montre d'une manière éclatante l'inanité des descriptions dichotomiques de ces deux manifestations de la vie.

C'est l'heuristique 2 qui a été choisie pour le programme démonstratif ci-après, mais ce choix est arbitraire. Le lecteur pourra facilement faire

la comparaison avec l'heuristique 1 en modifiant les lignes correspondantes.

Il est intéressant de constater, à ce stade de réflexion, qu'*il n'y a aucun rapport entre les performances d'une heuristique et sa complexité*, à moins que l'on ne donne justement un nom évocateur et peut-être mesurable à ce rapport : *la quantité d'intelligence contenue dans l'heuristique*.

Il y a peut-être là une piste pour définir et mesurer l'intelligence artificielle, comme on mesure depuis quelques décennies la quantité d'information contenue dans un message. Or, un programme est un message.

L'intelligence d'une heuristique serait alors mesurée par le rapport performance du programme/quantité d'information du programme. La notion de performance reste à préciser. Elle me semble liée à la *quantité d'information sécrétée par le programme en question.* Dans cette acception, l'intelligence d'une heuristique est un rapport sans dimension analogue à un coefficient d'amplification.

Plus un programme est court, plus il est intelligent à performance égale. Les prochains couples d'exemples E10-E11 ou E28-E29 nous fourniront de la matière complémentaire en ce domaine, de façon très concrète.

Je noterai, pour finir, que la relation de définition de la quantité d'I.A. amorcée ci-avant n'est peut-être pas linéaire. Les sensations sont en général logarithmiques vis-à-vis de leurs causes (loi de Weber-Fechner). Il pourrait en être de même pour la sensation d'intelligence que nous procure une heuristique. Nous y reviendrons plus loin à la lumière d'autres exemples.

En tout état de cause, nous poursuivrons l'investissement de la forteresse esprit, en amateurs bien sûr.

Programme

```
100 'ETAPE E5:MISSILE AUTO-GUIDE,CIBLE m
obile,TERRAIN NU,HEURISTIQUE DE NAVIGATI
ON (2);COURBE DITE "DU CHIEN"
110 INK 0,22:INK 1,1:INK 2,6:INK 3,23:CL
S:CLEAR
120 SYMBOL AFTER 32 :SYMBOL 33,&F0,&E0,&
F0,&B8,&1C,&F,&6,&4
130 'NOMBRES ALEATOIRES VARIES
140 RANDOMIZE TIME
150 CLS:H=INT(2*RND):IF H=1 THEN PXC=INT
(10*RND+3):PYC=3 ELSE PXC=3:PYC=INT(16*R
ND+8)'TIRAGE POSITION CIBLE
160 PPXC=PXC :PPYC=PYC
170 PXMI=39:PYMI=INT(20*RND+3):PPXMI=PXM
I:PPYMI=PYMI'POS DEP MISSILE ET MISE EN
RES
```

```
180 PEN 1:LOCATE PXMI,PYMI:PRINT CHR$(33
):PEN 2:LOCATE PXC,PYC:PRINT "o"'AFF MIS
SILE ET CIBLE
190 VXC=INT(3*RND)-1 :IF VXC=0 THEN VYC=
1 ELSE VYC=INT(3*RND-1) 'TIRAGE COMP VIT
ESSE CIBLE
200 SOUND 1,200,80,7:FOR AT=1 TO 2000:NE
XT'TOP DEPART
1000 REM BOUCLE DES MOUVEMENTS
1010 FOR M= 1 TO 500
1020 REM CIBLE
1030 PXC=PXC+VXC :PYC=PYC+VYC 'CALC NOUV
ELLE POS
1040 GOSUB 3000' EFFETS DE BORDS
1050 PEN 2 :IF PXC<>40 OR PYC<>25 THEN L
OCATE PPXC,PPYC :PRINT " ":LOCATE PXC,PY
C :PRINT"o" :SOUND 1,100,5,7 :PPXC=PXC :
PPYC=PYC ELSE SOUND 1,53,10,7 :PXC=39 :V
XC=-ABS(VXC) :VYC=-ABS(VYC) :GOTO 1050 '
AFF CIBLE ET MISE EN RES
1060 GOSUB 4000'CIBLE ATTEINTE
1070 REM MISSILE:HEURIST NAV (2)
1080 VXMI=SGN(PXC-PXMI):VYMI=SGN(PYC-PYM
I)
1090 PXMI=PPXMI+VXMI:PYMI=PPYMI+VYMI'CAL
C NOUV POS
1100 PEN 3:LOCATE PPXMI,PPYMI:PRINT CHR$
(33):PEN 1:LOCATE PXMI,PYMI:PRINT CHR$(3
3):SOUND 1,30,1,7:PPXMI=PXMI:PPYMI=PYMI'
AFF ET MISE EN RES
1110 GOSUB 4000'CIBLE ATTEINTE
1120 FOR AT=1 TO 50:NEXT:NEXT M
1130 END
2000 REM ZONE DES SOUS-PROGRAMMES
3000 REM SSPROG EFFETS DE BORDS CIBLE
3010 IF PXC >0 THEN 3020 ELSE SOUND 1,53
,10,7 :PXC=1 :VXC=-VXC
3020 IF PXC< 41 THEN 3030 ELSE SOUND 1,5
3,10,7 :PXC=40 :VXC=-VXC
3030 IF PYC>0 THEN 3040 ELSE SOUND 1,53,
10,7 :PYC=1 :VYC=-VYC :RETURN
3040 IF PYC<26 THEN RETURN ELSE SOUND 1,
53,10,7 :PYC=25 :VYC=-VYC :RETURN
4000 REM SSPROG CIBLE ATTEINTE
4010 IF PXMI<>PXC THEN RETURN ELSE IF PY
MI<>PYC THEN RETURN
4020 FOR FOIS=1 TO 4:SOUND 1,27,5,7:FOR
AT=1 TO 100:NEXT:NEXT:FOR AT=1 TO 500:NE
XT:GOTO 110
```

Commentaires techniques

- La vitesse de la cible, uniforme tout au cours de la poursuite, est définie en amont de la boucle des mouvements. Ses deux composantes valent 1, 0 ou – 1 afin d'engendrer des mouvements de directions variées dont chaque pas franchisse une case d'écran.
- La boucle des mouvements est commune à la cible et au missile.
- Il y a, comme précédemment, deux sous-programmes, l'un pour les effets de bords, l'autre pour surveiller l'impact du missile sur la cible.

ÉTAPE E6

UNE HEURISTIQUE DE SURVIE

Présentation

Nous revenons au missile autoguidé de E4 avec sa cible fixe, mais une autre complication surgit : le terrain se hérisse de 31 obstacles (des piliers) qui en font un véritable labyrinthe.

Il faut préciser que le missile est, comme précédemment, doté de la vue pour apprécier la direction de la cible, mais qu'il ne voit pas les piliers ; il n'en détecte la présence qu'en les touchant.

On peut supposer par exemple qu'il fait nuit et que la cible est lumineuse, ou bien qu'il s'agit d'un labyrinthe dont les piliers sont transparents, ou tout autre décor possédant le même isomorphisme sensoriel, à savoir : détection directionnelle de la cible, détection par contact des piliers.

L'heuristique, c'est-à-dire le fil de pensée pratique, adoptée dans le présent exemple, se compose de deux heuristiques distinctes :

- l'heuristique de navigation 1 déjà examinée, valable en terrain nu et consistant, en l'absence d'obstacle, à se rapprocher plus ou moins de la cible ;
- une heuristique de dégagement repérée A qui permet de contourner les piliers rencontrés, de neutraliser en quelque sorte ces intrus.

Il faut bien voir que l'opportunité d'une telle décomposition n'est ni naturelle, ni évidente. En effet, pourquoi naviguer comme s'il n'y avait pas de piliers puis, au moment où l'on arrive dessus, se décider à réagir, alors qu'il serait peut-être plus payant de les éviter préventivement en faisant un détour ?

Niels Bohr, dans sa géniale simplicité, a proposé une réponse : il est délicat de faire des prévisions, surtout pour l'avenir...

Cette boutade nous conduit tout de même à préciser la règle du jeu, pierre d'achoppement permanente de la robotique : le missile ignore au départ la position des piliers et, même pour les parcours suivants, il n'en conserve pas la mémoire. Nous verrons qu'il n'en sera pas toujours ainsi, mais il faudra alors le dire.

Ainsi équipé, le missile fait jouer à plein le principe de la fatigue mini-

male : "Si je me rapproche mine de rien de la cible, cela va dans le bon sens, et si un pilier se met en travers de ma route je fais semblant d'aller ailleurs puis, mine de rien, je me rapproche de la cible..." Cela admis, le problème sera réglé lorsqu'on aura choisi la tactique de contournement des piliers.

Le principal est de se dégager coûte que coûte et il s'agit en fait de survivre pour continuer. Si le missile ne se dégage pas, la superbe heuristique de navigation sombrera dans l'inutile car il n'y aura plus de navigation.

Si l'on compare la mission du missile à une guerre, on peut dire que le but en est défini par le chef de l'État, que l'heuristique de navigation est l'affaire du général et que l'heuristique de dégagement repose sur les épaules du simple troufion (ce mot existe heureusement dans le dictionnaire car il illustre parfaitement la progression des rôles).

Quelle est la plus noble des deux heuristiques ? Vous aurez cent réponses de cent personnes.

Si la noblesse consiste à mépriser les détails subalternes, c'est le général et son heuristique de navigation qui dominent largement. L'idée philosophique de base est, dans ce cas, qu'on ne peut atteindre un objet avant de s'en être d'abord rapproché.

Si la noblesse consiste à vouloir remplir la mission, gloire aux soldats et aux heuristiques de dégagement ! L'idée philosophique de base est alors du même degré de subtilité que la précédente : pour aller quelque part, il faut pouvoir bouger.

La mère de famille aura des opinions différentes suivant que son fils est adulte ou nourrisson et, dans ce dernier cas, suivant la carrière qu'elle ambitionne pour lui : général ou caporal, jamais soldat.

D'un point de vue bassement pratique, contentons-nous d'observer que, sans heuristique de dégagement, la partie est perdue d'avance alors qu'elle peut se passer d'heuristique de navigation comme nous l'avons vu en E2.

Admettons cependant qu'en plus de nombreux soldats, il faille quelques colonels.

Revenons à nos piliers. L'heuristique de dégagement A choisie consiste, lorsque le missile touche un pilier, à se placer dans la case nord-est contiguë à celle occupée précédemment ; le missile se dégage donc systématiquement par le nord-est.

On constate deux choses avec cette méthode :

- Elle est d'un déterminisme absolu. Associée à une heuristique de navigation donnée (heuristique 1 en l'occurrence), elle mène le missile par un itinéraire unique.

- Elle ne réussit pas toujours le dégagement. En effet, si un autre pilier se trouve par hasard à la position convoitée, le missile est définitivement bloqué.

Nous n'avons choisi cette heuristique A qu'à titre de contre-exemple. Un humain ne tomberait jamais dans une telle chausse-trappe et changerait vite d'heuristique après s'être cogné la tête une ou deux fois.

Programme

```
100 'ETAPE E6:MISSILE AUTO-GUIDE,CIBLE F
IXE,31 OBSTACLES,HEURISTIQUE DE NAVIGATI
ON (1),HEURISTIQUE DE DEGAGEMENT (A),3 P
ARCOURS SUR MEME TERRAIN
110 INK 0,16:INK 1,1:INK 2,6:INK 3,21:CL
S:CLEAR
120 DIM PXPI(30),PYPI(30)'LISTE DES COOR
DONNEES DES 30 PILIERS
130 SYMBOL AFTER 32:SYMBOL 33,&F0,&E0,&F
0,&B8,&1C,&F,&6,&4
140 'NOMBRES ALEATOIRES VARIES
150 RANDOMIZE TIME
160 RG=1'RANG DU PARCOURS
170 CLS:H=INT(2*RND):IF H=1 THEN PXC=INT
(25*RND+5):PYC=6 ELSE PXC=6:PYC=INT(12*R
ND+5)'TIRAGE POSITION CIBLE
180 PXMI=36:PYMI=21'POSITION DEPART MISI
LE
190 PPXMI=PXMI:PPYMI=PYMI'MISE EN RES PO
UR TRACE
200 PEN 1:LOCATE PXMI,PYMI:PRINT CHR$(33
):PEN 2:LOCATE PXC,PYC:PRINT "o"'AFF MIS
SILE ET CIBLE
210 IF RG>1 THEN 280
220 FOR N=0 TO 30:PXPI(N)=INT((33-PXC)*R
ND)+PXC-1:PYPI(N)=INT((21-PYC)/(36-PXC)*
(PXPI(N)-PXC))+INT((5*RND-2)*(PXPI(N)-PX
C)/8)+PYC+1
230 IF PYPI(N)>PYC-2 THEN 250
240 PYPI(N)=PYC-1:GOTO 270
250 IF PYPI(N)<>PYC THEN IF PYPI(N)<22 T
HEN 270 ELSE PYPI(N)=21:GOTO 270
260 IF PXPI(N)<>PXC THEN 270 ELSE PYPI(N
=PYC-1
270 PEN 1:LOCATE PXPI(N),PYPI(N):PRINT"=
":NEXT'AFF PILIER
280 SOUND 1,200,80,7:FOR AT=1 TO 2000:NE
XT'TOP DEPART
1000 REM BOUCLE DES MOUVEMENTS
1010 FOR M=1 TO 50'50 MOUVMTS MAXI ADMIS
 POUR ATENDRE CIBLE
1020 REM MISSILE,HEURIST NAV (1
```

```
1030 IF PXMI<>PXC THEN NUME=ABS(PYMI-PYC
):DENO=ABS(PXMI-PXC):IN=NUME/DENO ELSE 1
060
1040 IF IN<0.5 THEN VXMI=SGN(PXC-PXMI):V
YMI=0:GOTO 1070
1050 IF IN>2 THEN VXMI=0:VYMI=SGN(PYC-PY
MI):GOTO 1070
1060 VXMI=SGN(PXC-PXMI):VYMI=SGN(PYC-PYM
I)
1070 TT=0'INIT RANG DE PASS DS SSPROG HE
URT
1080 PXMI=PPXMI+VXMI:PYMI=PPYMI+VYMI.
1090 IF PXMI>31 THEN 1110 ELSE GOSUB 300
0'TEST HEURT PILIER
1100 IF OBS=1 THEN 1080
1110 PEN 3:LOCATE PPXMI,PPYMI:PRINT CHR$
(33):PEN 1:LOCATE PXMI,PYMI:PRINT CHR$(3
3):SOUND 1,30,5,7:PPXMI=PXMI:PPYMI=PYMI
1120 GOSUB 4000'TEST CIBLE ATTEINTE
1130 FOR N=1 TO 80:NEXT:NEXT M
1140 SOUND 1,500,40,7:FOR AT=1 TO 800:NE
XT:GOTO 110'PARCOURS RATE
1150 END
2000 REM ZONE DES SOUS-PROGRAMMES
3000 REM SSPROG HEURT PILIER
3010 OBS=0'NEUTRAL HEURT PRECEDENT
3020 FOR N=0 TO 30:IF PXMI<>PXPI(N) THEN
 3060 ELSE IF PYMI<>PYPI(N) THEN 3060
3030 FOR AT=1 TO 150:NEXT:SOUND 1,24,5,7
::PEN 2:LOCATE PXMI,PYMI:PRINT"*":FOR AT
=1 TO 50:NEXT:PEN 1:LOCATE PXMI,PYMI:PRI
NT"="
3040 VXMI=1:VYMI=-1'HEURIST DE DEGAGEMEN
T (A)
3050 N=30:OBS=1'SORTIE BOUCLE DE RECHERC
HE DES OBSTACLES ET SIGNALMT OBSTACLE
3060 NEXT N
3070 FOR AT=1 TO 50:NEXT:TT=TT+1:IF TT<6
 THEN RETURN ELSE FOR AT=1 TO 800:NEXT:G
OTO 110'CAS OU LE MISSILE NE TROUVE PAS
DE PLACE LIBRE
4000 REM SSPROG CIBLE ATTEINTE
4010 IF PXMI<>PXC THEN RETURN ELSE IF PY
MI<>PYC THEN RETURN
4020 FOR FOIS=1 TO 4:SOUND 1,27,5,7:FOR
AT=1 TO 100:NEXT:NEXT:FOR AT=1 TO 800:NE
XT
4030 RG=RG+1:IF RG<4 THEN PEN 3:LOCATE P
PXMI,PPYMI:PRINT CHR$(33):GOTO 180 ELSE
110
```

Commentaires techniques

- Les piliers sont volontairement placés dans la zone du parcours que doit emprunter le missile, avec une marge aléatoire de dispersion. C'est ce qui explique la forme curieuse de l'expression mathématique qui donne la valeur de l'ordonnée PYPI des piliers ; il est assez facile de voir que la relation entre PXPI et PYPI est une fonction linéaire, donc une droite passant par la cible et par le point de départ du missile, assortie d'une certaine marge et de quelques garde-fous complémentaires.
- On a choisi, comme dans la plupart des programmes qui suivront, l'heuristique de navigation 1, bien qu'elle ne soit pas la plus simple. Ce choix est arbitraire et facilement modifiable.
- Dans le sous-programme obstacle, on limite à 5 le nombre de tentatives infructueuses de dégagement (variable TT) puisque toute tentative du même genre est vouée à l'échec.

ÉTAPE E7

VIVENT LES ROBOTS !

Présentation

Nous arrivons au premier exemple digne, à nos yeux, de mériter le label de l'I.A., un petit label d'amateur dans lequel notre projectile larve, devenu missile chrysalide dans le cocon E4-E5-E6, apparaît maintenant dans sa livrée de robot libellule.

C'est le choix décisif d'une heuristique de dégagement appropriée qui va lui conférer ce standing et qui lui permettra, au fil des exemples suivants, d'en mériter les grades supérieurs. Appelons-la heuristique B.

La simple idée de base est que, puisque le robot, au moment où il touche un pilier, ignore où se trouve les autres obstacles du voisinage, il doit essayer une autre case puis, en cas d'insuccès, une autre, etc. La manière la moins fatigante de le faire consiste à tâter une case au hasard, puis une autre au hasard, jusqu'à trouver une case libre.

Le résultat est assez spectaculaire et ce système de bas étage réussit toujours à mener notre robot à son maître dans la limite réglementaire des 50 pas maximaux admis.

J'ai omis de préciser qu'il n'y a plus de cible, mais un maître aimant son robot et adoré de lui, le second ne songeant qu'à rejoindre le premier.

Outre son efficacité, essentiellement due à l'élimination de tout blocage, cet exemple illustre la possibilité de choisir plusieurs attitudes et d'introduire du libre arbitre grâce à une ignorance partielle des données (la présence et la position d'éventuels piliers dans le voisinage).

Le programme suivant répète trois parcours sur le même terrain et l'on voit notre robot hésiter, tâtonner, revenir parfois sur ses pas, et partir par des chemins divers. Ce robot décontracté perdra de sa spontanéité plus tard, au fur et à mesure que ses facultés se développeront.

Pour l'instant, accordons-lui le diplôme de robot, car il possède maintenant bon nombre des critères significatifs du Chapitre III si l'on excepte *a*, *b*, *j*, *k* et *l* qui apparaîtront ultérieurement.

Programme

```
100 'ETAPE E7:1ER PROGRAMME DE ROBOT;ROB
OT CHERCHANT SON MAITRE DANS LABYRINTHE.
```

```
31 PILIERS,HEURISTIQUE DE NAVIGATION (1)
,HEURISTIQUE DE DEGAGEMENT (B),3 PARCOUR
S SUR MEME TERRAIN
110 INK 0,16:INK 1,1:INK 2,6:INK 3,21:CL
S:CLEAR
120 DIM PXPI(30),PYPI(30)'LISTE DES COOR
DONNEES DES 31 PILIERS
130 SYMBOL AFTER 32:SYMBOL 33,&0,&7C,&D6
,&6C,&38,&FE,&7C,&6C'DESSIN ROBOT;DESSIN
 MAITRE:CODE 250
140 'NOMBRES ALEATOIRES VARIES
150 RANDOMIZE TIME
160 RG=1'RANG DU PARCOURS
170 CLS:H=INT(2*RND):IF H=1 THEN PXM=INT
(25*RND+5):PYM=6 ELSE PXM=6:PYM=INT(12*R
ND+5)'TIRAGE POSITION MAITRE
180 PXR=36:PYR=21'POSITION DEPART ROBOT
190 PPXR=PXR:PPYR=PYR 'MISE EN RES POUR
TRACE
200 PEN 1:LOCATE PXR,PYR:PRINT CHR$(33):
PEN 2:LOCATE PXM,PYM:PRINT CHR$(250)'AFF
 ROBOT ET MAITRE
210 IF RG>1 THEN 280
220 FOR N=0 TO 30:PXPI(N)=INT((33-PXM)*R
ND)+PXM-1:PYPI(N)=INT((21-PYM)/(36-PXM)*
(PXPI(N)-PXM))+INT((5*RND-2)*(PXPI(N)-PX
M)/8)+PYM+1
230 IF PYPI(N)>PYM-2 THEN 250
240 PYPI(N)=PYM-1:GOTO 270
250 IF PYPI(N)<>PYM THEN IF PYPI(N)<22 T
HEN 270 ELSE PYPI(N)=21:GOTO 270
260 IF PXPI(N)<>PXM THEN 270 ELSE PYPI(N
=PYM-1
270 PEN 1:LOCATE PXPI(N),PYPI(N):PRINT"=
":NEXT'AFF PILIER
280 SOUND 1,200,80,7:FOR AT=1 TO 2000:NE
XT'TOP DEPART
1000 REM BOUCLE DES MOUVEMENTS
1010 FOR M=1 TO 50'50 MOUVMTS MAXI ADMIS
 POUR ATTEINDRE MAITRE
1020 REM HEURIST NAV (1)
1030 IF PXR<>PXM THEN NUME=ABS(PYR-PYM):
DENO=ABS(PXR-PXM):IN=NUME/DENO ELSE 1060
1040 IF IN<0.5 THEN VXR=SGN(PXM-PXR):VYR
=0:GOTO 1070
1050 IF IN>2 THEN VXR=0:VYR=SGN(PYM-PYR)
:GOTO 1070
1060 VXR=SGN(PXM-PXR):VYR=SGN(PYM-PYR)
1070 PXR=PPXR+VXR:PYR=PPYR+VYR
1080 IF PXR>31 THEN 1100 ELSE GOSUB 3000
'TEST HEURT PILIER
1090 IF OBS=1 THEN 1070
1100 PEN 3:LOCATE PPXR,PPYR:PRINT CHR$(3
3):PEN 1:LOCATE PXR,PYR:PRINT CHR$(33):S
```

```
OUND 1,30,5,7:PPXR=PXR:PPYR=PYR
1110 GOSUB 4000'TEST MAITRE REJOINT
1120 FOR AT=1 TO 80:NEXT:NEXT M
1130 SOUND 1,500,40,7:GOTO 4030'PARCOURS
 RATE
1140 END
2000 REM ZONE DES SOUS-PROGRAMMES
3000 REM SSPROG HEURT PILIER
3010 OBS=0'NEUTRAL HEURT PRECEDENT
3020 FOR N=0 TO 30:IF PXR<>PXPI(N) THEN
3080 ELSE IF PYR<>PYPI(N) THEN 3080
3030 FOR AT=1 TO 150:NEXT:SOUND 1,24,5,7
::PEN 2:LOCATE PXR,PYR:PRINT"*":FOR AT=1
 TO 50:NEXT:PEN 1:LOCATE PXR,PYR:PRINT"=
"
3040 REM HEURIST DE DEGAGMT (B)
3050 S1=INT(3*RND)-1 :IF S1<>0 THEN S2=I
NT(3*RND)-1 ELSE S2=INT(2*RND)-0.5
3060 VXR=SGN(S1) :VYR=SGN(S2)
3070 N=30 :OBS=1 'SORTIE BOUCLE DE RECHE
RCHE DES OBSTACLES ET SIGNALMT OBSTACLE
3080 NEXT N
3090 RETURN
4000 REM SSPROG MAITRE REJOINT
4010 IF PXR<>PXM THEN RETURN ELSE IF PYR
<>PYM THEN RETURN
4020 FOR FOIS=1 TO 4:SOUND 1,27,5,7:FOR
AT=1 TO 100:NEXT:NEXT
4030 FOR AT=1 TO 800:NEXT:RG=RG+1:IF RG<
4 THEN PEN 3:LOCATE PPXR,PPYR:PRINT CHR$
(33):GOTO 180 ELSE 110
```

Commentaires techniques

Dans le sous-programme heurt pilier, la direction de dégagement est tirée au sort par le relais des deux variables S1 et S2, susceptibles de prendre une valeur positive, négative ou nulle et fixant ainsi la valeur des composantes VX et VY à 1, – 1 ou 0, permettant donc d'atteindre suivant les cas l'une des huit positions contiguës.

V

À LA RECHERCHE D'UNE MARTINGALE

"Puisque ces mystères me dépassent, feignons d'en être l'organisateur."

J. COCTEAU.

Jouer aux dés pour trouver la sortie, telle est l'essence de l'heuristique de dégagement B de l'étape précédente.

Mais il y a de multiples façons de jouer aux dés, y compris en les pipant pour faire sortir plus souvent les bons numéros. C'est à cette œuvre de longue haleine que notre robot va progressivement se consacrer dans le présent chapitre, de E8 à E13.

Il va d'abord essayer de dénicher une martingale infaillible, comme le sont toutes les martingales avant qu'on les essaie, puis, déçu par l'indifférence constante du dieu Hasard, il perdra la foi et se décidera... à décider lui-même.

Nous verrons ainsi successivement à l'œuvre les heuristiques de dégagement C, D, E et F, en attendant mieux. La technicité des programmes de démonstration en est progressivement accrue, car les facultés sensorielles ou intellectuelles supplémentaires requises, en particulier les fonctions de mémorisation et de comparaison, se traduisent par des manipulations supplémentaires d'informations.

Avant de poursuivre le chemin ainsi tracé, faisons rapidement le point de ce que nous pouvons maintenant dire, concrètement, des heuristiques introduites aux étapes E4 et E6.

L'heuristique de *navigation* permet d'*élaguer l'arbre* très touffu *des mouvements possibles*, en ne retenant que les meilleurs, *abstraction faite des obstacles* : c'est une *stratégie* d'*attaque*.

L'heuristique de dégagement permet de *traiter les obstacles* de la meilleure façon : c'est une *tactique* de *défense*.

C'est à dessein que les mots caractéristiques ont été mis en relief pour marquer, non pas leur opposition, mais leur complémentarité. Cette clarification des idées, ainsi que la terminologie choisie pour les exprimer, me semble aller bien au-delà des préoccupations ludiques de nos petits bonshommes [1].

1. Jacques Pitrat a développé le premier, en termes différents, et à propos du jeu d'échecs, l'idée de stratégie élagueuse : *Journal of Artificial Intelligence*, vol. 8, n° 3, juin 1977.

ÉTAPE E8

UN PEU D'ORDRE

Présentation

Avec l'heuristique de dégagement B de E7, le robot misait totalement sur le hasard pour trouver une case libre parmi les huit à portée de son pied. Il pouvait tomber sur une telle case du premier coup ou faire de très nombreuses tentatives, 5, 10, ou même 100 avant d'y parvenir. Avec un peu de patience, on assiste effectivement à des sarabandes de ce genre sur l'écran.

Cette fois, il fait un effort d'organisation (le moins fatigant des efforts intellectuels) en tirant la première direction au hasard puis, en cas d'insuccès, en explorant les sept autres positions dans l'ordre, soit dans le sens des aiguilles d'une montre normalement constituée, soit dans le sens inverse, ce sens de rotation étant lui-même tiré au hasard. On obtient ainsi l'heuristique de dégagement C.

L'indéterminisme du trajet subsiste intégralement puisque la première direction est tirée au sort ainsi que le sens d'exploration. On peut constater, à l'exécution du programme suivant, que les trois parcours successivement effectués sont aussi variés que dans E7.

Mais il y a une différence essentielle : le nombre maximal de tentatives infructueuses est réduit à 7 et, moins évident, le nombre *moyen* de tentatives infructueuses est également plus faible, car tout risque de répétition est exclu. Les trajets sont donc plus rapides qu'en E7.

Cette heuristique implique, en contrepartie, que le robot soit doué de mémoire pour organiser son exploration circulaire. Bien entendu, le programme démonstratif doit, lui aussi, simuler cette mémorisation.

Programme

```
100 'ETAPE E8:ROBOT CHERCHANT SON MAITRE
 DANS LABYRINTHE,31 PILIERS,HEURISTIQUE
DE NAVIGATION (1),HEURISTIQUE DE DEGAGEM
ENT (C),4 PARCOURS SUR MEME TERRAIN
110 INK 0,16:INK 1,1:INK 2,6:INK 3,21:CL
S:CLEAR
120 DIM PXPI(30),PYPI(30)'LISTE DES COOR
DONNEES DES 31 PILIERS
```

```
130 SYMBOL AFTER 32:SYMBOL 33,&0,&7C,&D6
,&6C,&38,&FE,&7C,&6C'DESSIN ROBOT;DESSIN
 MAITRE:CODE 250
140 'NOMBRES ALEATOIRES VARIES
150 RANDOMIZE TIME
160 RG=1'RANG DU PARCOURS
170 CLS:H=INT(2*RND):IF H=1 THEN PXM=INT
(25*RND+6):PYM=6 ELSE PXM=6:PYM=INT(12*R
ND+6)'TIRAGE POSITION MAITRE
180 PXR=36:PYR=21'POSITION DEPART ROBOT
190 PPXR=PXR:PPYR=PYR 'MISE EN RES POUR
TRACE
200 PEN 1:LOCATE PXR,PYR:PRINT CHR$(33):
PEN 2:LOCATE PXM,PYM:PRINT CHR$(250)'AFF
 ROBOT ET MAITRE
210 IF RG>1 THEN 280
220 FOR N=0 TO 30:PXPI(N)=INT((33-PXM)*R
ND)+PXM-1:PYPI(N)=INT((21-PYM)/(36-PXM)*
(PXPI(N)-PXM))+INT((5*RND-2)*(PXPI(N)-PX
M)/8)+PYM+1
230 IF PYPI(N)>PYM-2 THEN 250
240 PYPI(N)=PYM-1:GOTO 270
250 IF PYPI(N)<>PYM THEN IF PYPI(N)<22 T
HEN 270 ELSE PYPI(N)=21:GOTO 270
260 IF PXPI(N)<>PXM THEN 270 ELSE PYPI(N
=PYM-1
270 PEN 1:LOCATE PXPI(N),PYPI(N):PRINT"=
":NEXT'AFF PILIER
280 SOUND 1,200,80,7:FOR AT=1 TO 2000:NE
XT'TOP DEPART
1000 REM BOUCLE DES MOUVEMENTS
1010 FOR M=1 TO 50'50 MOUVMTS MAXI ADMIS
 POUR ATTEINDRE MAITRE
1020 REM HEURIST NAV (1)
1030 IF PXR<>PXM THEN NUME=ABS(PYR-PYM):
DENO=ABS(PXR-PXM):IN=NUME/DENO ELSE 1060

1040 IF IN<0.5 THEN VXR=SGN(PXM-PXR):VYR
=0:GOTO 1080
1050 IF IN>2 THEN VXR=0:VYR=SGN(PYM-PYR)
:GOTO 1080
1060 VXR=SGN(PXM-PXR):VYR=SGN(PYM-PYR)
1070 TT=0'INITIALISATION DU RANG TT DES
TENTATIVES POUR TROUVER UNE POSITION LIB
RE APRES AVOIR HEURTE UN PILIER
1080 PXR=PPXR+VXR:PYR=PPYR+VYR
1090 IF PXR>31 THEN 1110 ELSE GOSUB 3000
'TEST HEURT PILIER
1100 IF OBS=1 THEN 1080
1110 PEN 3:LOCATE PPXR,PPYR:PRINT CHR$(3
3):PEN 1:LOCATE PXR,PYR:PRINT CHR$(33):S
OUND 1,30,5,7:PPXR=PXR:PPYR=PYR
1120 GOSUB 4000'TEST MAITRE REJOINT
1130 FOR N=1 TO 80:NEXT:NEXT M
```

```
1140 SOUND 1,500,40,7:GOTO 4030'PARCOURS
 RATE
1150 END
2000 REM ZONE DES SOUS-PROGRAMMES
3000 REM SSPROG HEURT PILIER
3010 OBS=0:TT=TT+1
3020 FOR N=0 TO 30:IF PXR<>PXPI(N) THEN
3190 ELSE IF PYR<>PYPI(N) THEN 3190
3030 FOR AT=1 TO 150:NEXT:SOUND 1,24,5,7
::PEN 2:LOCATE PXR,PYR:PRINT"*":FOR AT=1
 TO 50:NEXT:PEN 1:LOCATE PXR,PYR:PRINT"=
"
3040 REM HEURIST DE DEGAGMT (C):TIRAGE A
U SORT 1ERE DIRECTION DE DEGAGEMENT,PUIS
,EN CAS D'INSUCCES,EXPLORATION ORDONNEE
ES AUTRES DIRECTIONS
3050 IF TT=1 THEN HAS=INT(8*RND+1)::SS=I
NT(2*RND):GOTO 3090
3060 IF SS=0 THEN HAS=HAS+1::IF HAS<9 TH
EN 3090 ELSE HAS=HAS-8:GOTO 3090
3070 HAS=HAS-1:IF HAS>0 THEN 3090
3080 HAS=HAS+8
3090 IF HAS=1 THEN 3100 ELSE  IF HAS=2 T
HEN 3110 ELSE IF HAS=3 THEN 3120 ELSE IF
 HAS=4 THEN 3130 ELSE IF HAS=5 THEN 3140
 ELSE IF HAS=6 THEN 3150 ELSE IF HAS=7 T
HEN 3160 ELSE 3170
3100 VXR=1:VYR=1:GOTO 3180
3110 VXR=1:VYR=0:GOTO 3180
3120 VXR=1:VYR=-1:GOTO 3180
3130 VXR=0:VYR=-1:GOTO 3180
3140 VXR=-1:VYR=-1:GOTO 3180
3150 VXR=-1:VYR=0:GOTO 3180
3160 VXR=-1:VYR=1:GOTO 3180
3170 VXR=0:VYR=1
3180 N=30:OBS=1
3190 NEXT N
3200 RETURN
4000 REM SSPROG MAITRE REJOINT
4010 IF PXR<>PXM THEN RETURN ELSE IF PYR
<>PYM THEN RETURN
4020 FOR FOIS=1 TO 4:SOUND 1,27,5,7:FOR
AT=1 TO 100:NEXT:NEXT
4030 FOR AT=1 TO 800:NEXT:RG=RG+1:IF RG<
5 THEN PEN 3:LOCATE PPXR,PPYR:PRINT CHR$
(33):GOTO 180 ELSE 110
```

Commentaires techniques

- C'est le sous-programme heurt pilier qui est modifié par rapport à E7.
- Il faut un paramètre de comptage du rang de passage dans ce sous-programme à partir du moment où le robot se trouve au contact d'un

pilier. C'est le rôle de la variable TT qui est initialisée à 0 avant le premier calcul de position, puis est augmentée (incrémentée) de 1 à chaque passage dans le sous-programme.

Lorsqu'une position libre est trouvée, elle est occupée par le robot et, au prochain mouvement, c'est-à-dire à la prochaine boucle, le paramètre TT sera ramené à 0.

- Au premier passage (TT = 1), le sous-programme tire au sort la première direction testée parmi les huit possibles : c'est la variable HAS dont chaque valeur de 1 à 8 correspond à une orientation (1 = SE, 2 = E, 3 = NE, 4 = N, 5 = NO, 6 = O, 7 = SO, 8 = S). Il tire également le sens de rotation : variable SS à choisir entre 0 ou 1.
- Aux passages suivants (TT = 2, puis 3... si les tentatives restent infructueuses), HAS est augmentée de 1 ou diminuée de 1 suivant le sens d'exploration SS tiré au départ.
- À noter que si HAS ainsi modifiée est comprise entre 1 et 8 tout va bien, mais si HAS sort de ses limites on l'y ramène en lui ajoutant ou retranchant 8.

ÉTAPE E9

LA PUISSANCE DES PRÉDICATS

Présentation

La présente étape introduit une heuristique de dégagement D nettement plus évoluée que les précédentes ainsi qu'un nouveau mode de représentation des piliers dans le programme de simulation, utilisant la notion de *prédicat*, très prisée en I.A.

Examinons d'abord l'heuristique D, les prédicats étant eux-mêmes abordés dans les commentaires techniques du programme ci-après.

Le robot, au contact d'un pilier, opère comme en E8, mais en s'imposant une condition supplémentaire : il s'interdit de revenir, en se dégageant, sur la position qu'il occupait précédemment, sauf si aucune autre position n'est libre. Cette manière de penser est très naturelle puisqu'elle consiste à remarquer que, s'il y a un retour en arrière, au pas suivant le robot se retrouvera bêtement au même endroit, au contact du même pilier (pas de pèlerin ou pas de clerc, mais sûrement pas perdu).

Ce qui suppose que notre ami se souvienne de son dernier pas et que cela soit simulable. Ça l'est de plusieurs façons possibles, soit en mémorisant la position précédente (PPPX = PPX et PPPY = PPY), soit en mémorisant la vitesse du pas précédent (VVX = VX et VVY = VY). En E25 nous utiliserons une troisième méthode exploitant les fonctions logiques.

Pour l'instant, c'est la deuxième méthode qui sera adoptée, la plus simple en l'état actuel de notre progression.

La supériorité de l'heuristique D est très sensible ; les "tours de valse" constatés antérieurement, inadmissibles pour un esprit humain, sont presque éliminés, même si certains doubles retours en arrière, inexistants en E8, sont maintenant possibles.

L'indéterminisme est toujours présent, mais les parcours ont une autre allure, moins erratique.

En outre, la rapidité d'exécution due à l'utilisation des prédicats donne à Roméo — c'est ainsi que nous l'appellerons désormais — le pas alerte d'un conquistador, et cela malgré l'installation d'un frein dans le sous-programme de recherche d'une nouvelle direction, pour améliorer la perception des stades de réflexion du robot. Ce frein est classiquement réalisé par une boucle d'attente. Si le lecteur le supprime, les explorations sont menées à un train d'enfer qu'il est impossible de suivre.

Programme

```
100 'ETAPE E9:ROBOT CHERCHANT SON MAITRE
 DANS LABYRINTHE,31 PILIERS,HEURISTIQUE
DE NAVIGATION (1),HEURISTIQUE DE DEGAGEM
ENT (D),4 PARCOURS SUR MEME TERRAIN,NOUV
ELLE REPRESENTATION DES PILIERS

110 INK 0,19:INK 1,6:INK 2,1:INK 3,17:CL
S:CLEAR
120 DIM C(31,25)'TABLEAU DES COORDONNEES
 DES CASES DE L'ECRAN SUSCEPTIBLES DE RE
CEVOIR UN PILIER
130 SYMBOL AFTER 32:SYMBOL 33,&0,&7C,&D6
,&6C,&38,&FE,&7C,&6C'DESSIN ROBOT;DESSIN
 MAITRE:CODE 250
140 'NOMBRES ALEATOIRES VARIES
150 RANDOMIZE TIME
160 RG=1'RANG DU PARCOURS
170 CLS:H=INT(2*RND):IF H=1 THEN PXM=INT
(25*RND+6):PYM=6 ELSE PXM=6:PYM=INT(12*R
ND+6)'TIRAGE POSITION MAITRE
180 PXR=36:PYR=21:PPXR=PXR:PPYR=PYR'POSI
TION DEPART ROBOT ET MISE EN RESERVE POU
R TRACE
190 PEN 1:LOCATE PXR,PYR:PRINT CHR$(33):
PEN 2:LOCATE PXM,PYM:PRINT CHR$(250)'AFF
 ROBOT ET MAITRE
200 IF RG>1 THEN 1000
210 FOR N=0 TO 30:TERR$=INKEY$:IF TERR$<
>" " THEN 220 ELSE 110
220 I=INT((33-PXM)*RND)+PXM-1:J=INT((21-
PYM)/(36-PXM)*(I-PXM))+INT((5*RND-2)*(I-
PXM)/8)+PYM+1
230 IF J>PYM-2 THEN 250
240 J=PYM-1:GOTO 270
250 IF J<>PYM THEN IF J<22 THEN 270 ELSE
 J=21:GOTO 270
260 IF I<>PXM THEN 270 ELSE J=PYM-1
270 PEN 1:LOCATE I,J:PRINT"=":C(I,J)=1:N
EXT'AFF ET REPRESENTATION PILIER
280 SOUND 1,200,80,7:FOR AT=1 TO 2000:NE
XT'TOP DEPART
1000 REM BOUCLE DES MOUVEMENTS
1010 FOR M=1 TO 50'50 MOUVMTS MAXI ADMIS
 POUR ATTEINDRE MAITRE
1020 REM HEURIST NAV (1)
1030 IF PXR<>PXM THEN NUME=ABS(PYR-PYM):
DENO=ABS(PXR-PXM):IN=NUME/DENO ELSE 1060

1040 IF IN<0.5 THEN VXR=SGN(PXM-PXR):VYR
=0:GOTO 1070
1050 IF IN>2 THEN VXR=0:VYR=SGN(PYM-PYR)
:GOTO 1070
```

```
1060 VXR=SGN(PXM-PXR):VYR=SGN(PYM-PYR)
1070 TT=0'INITIALISATION DU RANG TT DES
TENTATIVES POUR TROUVER UNE POSITION LIB
RE APRES AVOIR HEURTE UN PILIER
1080 PXR=PPXR+VXR:PYR=PPYR+VYR
1090 IF PXR>31 THEN 1120 ELSE GOSUB 3000
'TEST HEURT PILIER
1100 IF OBS=0 THEN 1120 ELSE GOSUB 5000'
ESSAI NOUVELLE DIRECTION
1110 GOTO 1080'RECALCUL POSITION
1120 GOSUB 4000'TEST RETOUR EN ARRIERE
1130 IF RET=0 THEN 1150 ELSE GOSUB 5000'
ESSAI NOUV DIR
1140 GOTO 1080'RECALCUL NOUV POS
1150 PEN 3:LOCATE PPXR,PPYR:PRINT CHR$(3
3):PEN 1:LOCATE PXR,PYR:PRINT CHR$(33):S
OUND 1,30,5,7:PPXR=PXR:PPYR=PYR
1160 VVXR=VXR:VVYR=VYR'MEMO VITESE POUR
EVITER PROCHAIN RETOUR EN ARR
1170 GOSUB 6000'TEST MAITRE REJOINT
1180 FOR N=1 TO 80:NEXT:NEXT M
1190 SOUND 1,500,40,7:GOTO 6030'PARCOURS
 RATE
1200 END
2000 REM ZONE DES SOUS-PROGRAMMES
3000 REM SSPROG HEURT PILIER
3010 OBS=0
3020 IF C(PXR,PYR)<>1 THEN RETURN
3030 FOR AT=1 TO 150:NEXT:SOUND 1,24,5,7
:PEN 2:LOCATE PXR,PYR:PRINT"*":FOR AT=1
TO 50:NEXT:PEN 1:LOCATE PXR,PYR:PRINT"="
3040 OBS=1:TT=TT+1
3050 RETURN
4000 REM SSPROG RETOUR EN ARRIERE
4010 RET=0'NEUTRAL RET ARR PRECEDENT
4020 IF VXR<>-VVXR THEN RETURN ELSE IF V
YR<>-VVYR THEN RETURN
4030 IF TT>7 THEN RETURN ELSE FOR AT=1 T
O 100:NEXT AT:SOUND 1,20,5,7:PEN 2:LOCAT
E PXR,PYR:PRINT"*":FOR AT=1 TO 150:NEXT:
PEN 3:LOCATE PXR,PYR:PRINT CHR$(33):RET=
1:TT=TT+1:RETURN
5000 REM SSPROG ESSAI NOUVELLE DIRECTION
5010 REM HEURISTIQUE DE DEGAGEMENT (D):T
IRAGE 1ERE DIRECTION DE DEGAGMT PUIS,EN
CAS D'INSUCCES,EXPLOITATION ORDONNEE DES
 AUTRES DIRECIONS
5020 FOR AT=1 TO 100:NEXT AT:IF TT=1 THE
N HAS=INT(8*RND+1):SS=INT(2*RND):GOTO 50
60
5030 IF SS=0 THEN HAS=HAS+1::IF HAS<9 TH
EN 5060 ELSE HAS=HAS-8:GOTO 5060
5040 HAS=HAS-1:IF HAS>0 THEN 5060
5050 HAS=HAS+8
```

```
5060 IF HAS=1 THEN 5070 ELSE  IF HAS=2 T
HEN 5080 ELSE IF HAS=3 THEN 5090 ELSE IF
 HAS=4 THEN 5100 ELSE IF HAS=5 THEN 5110
 ELSE IF HAS=6 THEN 5120 ELSE IF HAS=7 T
HEN 5130 ELSE 5140
5070 VXR=1:VYR=1:GOTO 5150
5080 VXR=1:VYR=0:GOTO 5150
5090 VXR=1:VYR=-1:GOTO 5150
5100 VXR=0:VYR=-1:GOTO 5150
5110 VXR=-1:VYR=-1:GOTO 5150
5120 VXR=-1:VYR=0:GOTO 5150
5130 VXR=-1:VYR=1:GOTO 5150
5140 VXR=0:VYR=1
5150 RETURN
6000 REM SSPROG MAITRE REJOINT
6010 IF PXR<>PXM THEN RETURN ELSE IF PYR
<>PYM THEN RETURN
6020 FOR FOIS=1 TO 4:SOUND 1,27,5,7:FOR
AT=1 TO 100:NEXT:NEXT
6030 FOR AT=1 TO 800:NEXT:RG=RG+1:IF RG<
5 THEN PEN 3:LOCATE PPXR,PPYR:PRINT CHR$
(33):GOTO 180 ELSE 110
```

Commentaires techniques

- Les piliers ne sont plus représentés par leurs coordonnées PXPI(N) et PYPI(N) comme précédemment, mais par leurs coordonnées I, J ; leur rang N n'est pas mémorisé car Roméo se moque de connaître le numéro du pilier auquel il se heurte, il a seulement besoin de savoir si, à cette position, il y a ou non un pilier.

  Pour chaque pilier de coordonnées I, J ainsi tiré au sort, on définit une variable C(I, J) qui prend alors la valeur 1. La variable C(I, J) garde la valeur 0 pour toutes les cases d'écran de coordonnées I, J non occupées par un pilier.

  En fait, C(I, J) est donc une *proposition* de même nature que OBS, qui peut prendre l'une des deux valeurs *vrai* ou *faux*. Mais cette proposition est affectée de deux *arguments* I et J représentant les coordonnées de l'endroit pour lequel elle fournit son information.

  Une telle proposition variable, munie d'arguments localisateurs [2], prend le nom de *prédicat*. En somme, un prédicat est à une proposition ce qu'une variable est à une constante.

2. Ces arguments numériques pourraient tout aussi bien, mais le BASIC ne le permet pas, être des "mots" et représenter autre chose que des coordonnées ; par exemple, si I représente un pays et si J représente une ville, et si le prédicat CAP(I, J) est une proposition affirmant que J est la capitale de I, on a CAP(FRANCE, PARIS) = 1 et CAP(USA, BLOIS) = 0.

- Les cases de l'écran sont ainsi divisées en deux sous-ensembles : celles où il n'y a rien et celles où il y a un pilier.

  Le prédicat est un mode simple et puissant de représentation qualitative de l'univers du robot.

  Nous accéderons en E18, puis en E25 et E26, à une notion encore plus générale, celle de *fonction logique*.

- Dans le sous-programme heurt pilier, le prédicat C(I, J) est instantanément exploitable, sans recherche fastidieuse de la coïncidence successive de la position du robot avec celle de chacun des 30 piliers. En effet, le robot étant placé aux coordonnées PXR et PYR, il suffit d'examiner si le prédicat C(PXR, PYR) vaut 1 ou 0. Dans le premier cas le robot est au contact d'un pilier, sinon la route est libre.

  La simplicité du test est remarquable, ainsi que sa rapidité comme on peut en juger à l'exécution.

- Sous-programme retour en arrière : il examine si les composantes VXR et VYR de la vitesse du pas envisagé (déterminées par le sous-programme essai nouvelle direction) sont bien, comme exigé, *différentes* du symétrique des composantes VVXR et VVYR du pas précédent. Sinon, c'est qu'il y a retour en arrière. Le sous-programme examine alors quel est le rang TT de la tentative en cours. Si l'on en est à la huitième, on passe outre car le tour complet des explorations a été fait.

  Si par contre TT est inférieur à 8, on continue l'exploration pour trouver une position qui ne corresponde ni à un pilier, ni à un retour en arrière.

- Pour faciliter la programmation de cette gymnastique mentale, on a scindé le processus en trois sous-programmes :

  Sous-programme heurt pilier, déjà examiné ;

  Sous-programme retour en arrière, déjà examiné ;

  Sous-programme nouvelle direction.

  Chaque fois que l'un des deux premiers détecte une interdiction (présence d'un pilier ou retour en arrière), il renvoie au troisième qui déclenche l'exploration suivante et définit les composantes de la vitesse correspondante. La nouvelle position est alors recalculée et le cérémonial recommence jusqu'à détection d'une position valable.

  Le drapeau OBS des programmes précédents est complété, aux mêmes fins, par un drapeau RET.

  En plus de la visualisation habituelle, chaque tentative de retour en arrière intempestif est ponctuée d'un clic caractéristique pour permettre de suivre les opérations.

- Attention lors du renvoi à un nouveau terrain quand les trois parcours du terrain précédent sont terminés : il faut annuler les piliers du terrain précédent, soit en passant par une instruction de neutralisation générale des variables si elle existe (CLEAR), soit en ramenant à 0 les prédicats C(I, J) pour toutes les positions I, J de l'écran. Sinon, les anciens piliers subsisteraient dans leurs effets bien qu'ils ne fussent plus affichés à l'écran.

D'une manière générale, avec ce système de représentation, l'apparition d'un élément quelconque doit se concrétiser dans le programme de deux façons simultanées :

- visuellement par affichage sur l'écran ;
- sémantiquement par la valorisation à 1 du prédicat représentatif.

Inversement, toute disparition d'élément doit se concrétiser par son effacement visuel et par son effacement sémantique (mise à 0 du prédicat).

Nous y reviendrons dans le cas plus général de E25 et E26.

ÉTAPE E10

ROMÉO PASSE L'IMPASSE

Présentation

Cet exemple est seulement destiné, mais c'est important, à illustrer l'efficacité de l'heuristique de dégagement D précédente dans le cas difficile d'un chemin-piège en impasse.

Si le robot s'engouffre dans une telle nasse, les heuristiques antérieures sont impuissantes à l'en sortir, car l'heuristique de navigation, quelle qu'elle soit, tend à le ramener constamment au fond de l'impasse. Il est facile de voir que l'heuristique D résoud parfaitement ce problème.

Une fois sorti du mauvais pas, ou bien le hasard l'en écarte définitivement, ou bien Roméo refait joyeusement une plongée qu'il saura plus tard éviter.

Programme

```
100 'ETAPE E10:ILLUSTRATON DE E9 PAR LE
CAS PARTICULIER D'UNE IMPASSE
110 INK 0,19:INK 1,6:INK 2,1:INK 3,17:CL
S:CLEAR
120 DIM C(31,25)'TABLEAU DES COORDONNEES
 DES CASES DE L'ECRAN SUSCEPTIBLES DE RE
CEVOIR UN PILIER
130 SYMBOL AFTER 32:SYMBOL 33,&0,&7C,&D6
,&6C,&38,&FE,&7C,&6C'DESSIN ROBOT;DESSIN
 MAITRE:CODE 250
140 'NOMBRES ALEATOIRES VARIES
150 RANDOMIZE TIME
160 RG=1'RANG DU PARCOURS
170 CLS:PXM=7:PYM=INT(5*RND)+10'TIRAGE P
OSITION MAITRE
180 PXR=30:PYR=12:PPXR=PXR:PPYR=PYR'POSI
TION DEPART ROBOT ET MISE EN RESERVE POU
R TRACE
190 PEN 1:LOCATE PXR,PYR:PRINT CHR$(33):
PEN 2:LOCATE PXM,PYM:PRINT CHR$(250)'AFF
 ROBOT ET MAITRE
200 IF RG>1 THEN 1000
210 DATA 24,11,24,12,24,13,25,11,25,13,2
6,11,26,13,27,11,27,13,23,11,23,10,23,9,
24,9,23,13,23,14,23,15,24,15,22,10,21,10
```

```
,21,9,21,8,22,14,21,14,21,15,21,16,9,8,9
,10,9,12,9,14,9,16
220 DATA 8,8,8,9,8,10,8,12,8,14,8,15,8,1
6,12,8,12,9,13,9,13,10,14,10,15,10,15,14
,14,14,13,14,13,15,12,15,12,16
230 FOR N=1 TO 49:READ I,J:C(I,J)=1:PEN
1:LOCATE I,J:PRINT"=":NEXT'REPRESENTATIO
N ET AFF PILIER
240 SOUND 1,200,80,7:FOR AT=1 TO 2000:NE
XT'TOP DEPART
1000 REM BOUCLE DES MOUVEMENTS
1010 FOR M=1 TO 70'70 MOUVMTS MAXI ADMIS
 POUR ATTEINDRE MAITRE
1020 REM HEURIST NAV (1)
1030 IF PXR<>PXM THEN NUME=ABS(PYR-PYM):
DENO=ABS(PXR-PXM):IN=NUME/DENO ELSE 1060
1040 IF IN<0.5 THEN VXR=SGN(PXM-PXR):VYR
=0:GOTO 1070
1050 IF IN>2 THEN VXR=0:VYR=SGN(PYM-PYR)
:GOTO 1070
1060 VXR=SGN(PXM-PXR):VYR=SGN(PYM-PYR)
1070 TT=0'INITIALISATION DU RANG TT DES
TENTATIVES POUR TROUVER UNE POSITION LIB
RE APRES AVOIR HEURTE UN PILIER
1080 PXR=PPXR+VXR:PYR=PPYR+VYR
1090 IF PXR>29 THEN 1120 ELSE GOSUB 3000
'TEST HEURT PILIER
1100 IF OBS=0 THEN 1120 ELSE GOSUB 5000'
ESSAI NOUVELLE DIRECTION
1110 GOTO 1080'RECALCUL POSITION
1120 GOSUB 4000'TEST RETOUR EN ARRIERE
1130 IF RET=0 THEN 1150 ELSE GOSUB 5000'
ESSAI NOUV DIR
1140 GOTO 1080'RECALCUL NOUV POS
1150 PEN 3:LOCATE PPXR,PPYR:PRINT CHR$(3
3):PEN 1:LOCATE PXR,PYR:PRINT CHR$(33):S
OUND 1,30,5,7:PPXR=PXR:PPYR=PYR
1160 VVXR=VXR:VVYR=VYR'MEMO VITESE POUR
EVITER PROCHAIN RETOUR EN ARR
1170 GOSUB 6000'TEST MAITRE REJOINT
1180 FOR N=1 TO 80:NEXT:NEXT M
1190 SOUND 1,500,40,7:GOTO 6030'PARCOURS
 RATE
1200 END
2000 REM ZONE DES SOUS-PROGRAMMES
3000 REM SSPROG HEURT PILIER
3010 OBS=0
3020 IF C(PXR,PYR)<>1 THEN RETURN
3030 FOR AT=1 TO 150:NEXT:SOUND 1,24,5,7
:PEN 2:LOCATE PXR,PYR:PRINT"*":FOR AT=1
TO 50:NEXT:PEN 1:LOCATE PXR,PYR:PRINT"="
3040 OBS=1:TT=TT+1
3050 RETURN
4000 REM SSPROG RETOUR EN ARRIERE
```

```
4010 RET=0'NEUTRAL RET ARR PRECEDENT
4020 IF VXR<>-VVXR THEN RETURN ELSE IF V
YR<>-VVYR THEN RETURN
4030 IF TT>7 THEN RETURN ELSE FOR AT=1 T
O 100:NEXT AT:SOUND 1,20,5,7:PEN 2:LOCAT
E PXR,PYR:PRINT"*":FOR AT=1 TO 150:NEXT:
PEN 3:LOCATE PXR,PYR:PRINT CHR$(33):RET=
1:TT=TT+1:RETURN
5000 REM SSPROG ESSAI NOUVELLE DIRECTION
5010 REM HEURISTIQUE DE DEGAGEMENQ (D):T
IRAGE 1ERE DIRECTION DE DEGAGEMENT PUIS,
EN CAS D'INSUCCES,EXPLORATION ORDONNEE D
ES AUTRES DIRECTIONS
5020 FOR AT=1 TO 100:NEXT AT:IF TT=1 THE
N HAS=INT(8*RND+1):SS=INT(2*RND):GOTO 50
60
5030 IF SS=0 THEN HAS=HAS+1::IF HAS<9 TH
EN 5060 ELSE HAS=HAS-8:GOTO 5060
5040 HAS=HAS-1:IF HAS>0 THEN 5060
5050 HAS=HAS+8
5060 IF HAS=1 THEN 5070 ELSE  IF HAS=2 T
HEN 5080 ELSE IF HAS=3 THEN 5090 ELSE IF
 HAS=4 THEN 5100 ELSE IF HAS=5 THEN 5110
 ELSE IF HAS=6 THEN 5120 ELSE IF HAS=7 T
HEN 5130 ELSE 5140
5070 VXR=1:VYR=1:GOTO 5150
5080 VXR=1:VYR=0:GOTO 5150
5090 VXR=1:VYR=-1:GOTO 5150
5100 VXR=0:VYR=-1:GOTO 5150
5110 VXR=-1:VYR=-1:GOTO 5150
5120 VXR=-1:VYR=0:GOTO 5150
5130 VXR=-1:VYR=1:GOTO 5150
5140 VXR=0:VYR=1
5150 RETURN
6000 REM SSPROG MAITRE REJOINT
6010 IF PXR<>PXM THEN RETURN ELSE IF PYR
<>PYM THEN RETURN
6020 FOR FOIS=1 TO 4:SOUND 1,27,5,7:FOR
AT=1 TO 100:NEXT:NEXT
6030 FOR AT=1 TO 800:NEXT:RG=RG+1:IF RG<
5 THEN PEN 3:LOCATE PPXR,PPYR:PRINT CHR$
(33):GOTO 180 ELSE 110
```

Commentaires techniques

Les piliers ne sont plus tirés au hasard car la position de ces obstacles répond à un souci particulier de démonstration. Aussi les positions sont-elles définies dans des lignes de data.

ÉTAPE E11

ROMÉO SE TAIT

Présentation

Notre robot est, cette fois, soumis à la même épreuve d'impasse que précédemment et il utilise la même heuristique D.

La différence consiste dans la suppression du dispositif audiovisuel du jalonnement intellectuel de Roméo. Seuls subsistent les bips rythmant ses pas.

L'effet est inattendu : Roméo paraît plus intelligent qu'en E10. Ses processus mentaux sont devenus transparents, seuls ses actes apparaissent, comme dans la réalité du comportement humain.

L'homme n'est pas essentiellement un animal doué de la parole, mais plutôt à mon avis un animal doué de la faculté de rester muet et d'agir en silence, faculté peu maîtrisée car elle implique le tri des communications inutiles.

La pensée ne semble pas avoir été faite pour être exhibée, mais pour préparer des actes physiques efficaces. Elle ne demande à être décortiquée et autopsiée que dans certaines situations spéciales : enseignement, justification, adhésion à une action concertée. Écrire un livre représente également l'une de ces situations.

Mais, dans les cas normaux et spécialement dans les situations personnelles délicates, le silence est d'or, et l'une des dimensions de l'intelligence réside dans la discrétion avec laquelle elle fonctionne. L'esprit est l'un des moyens de compétition entre les humains, comme les biceps ou l'adresse manuelle, et sa vraie vocation est l'invisibilité, qu'on le déplore ou non.

Dans le prolongement des réflexions amorcées en E5 sur la mesurabilité de l'intelligence artificielle, on peut constater que le programme E11 est plus court que E10 puisqu'il a été élagué de tout son parasitisme indûment destiné à autrui. La sauce a été réduite mais le mets subsiste, concentré, affûté. Ses adhérences sociales en ralentissaient d'ailleurs l'exécution ; devenu secret, il sécrète davantage. Il a un coefficient d'amplification supérieur, il contient plus d'intelligence, même si la morale n'y trouve pas son compte.

Programme

```
100 'ETAPE E11:COMME E10 SANS SONO NI VI
SUALISATION
110 INK 0,19:INK 1,6:INK 2,1:INK 3,17:CL
S:CLEAR
120 DIM C(31,25)'TABLEAU DES COORDONNEES
 DES CASES DE L'ECRAN SUSCEPTIBLES DE RE
CEVOIR UN PILIER
130 SYMBOL AFTER 32:SYMBOL 33,&0,&7C,&D6
,&6C,&38,&FE,&7C,&6C'DESSIN ROBOT;DESSIN
 MAITRE:CODE 250
140 'NOMBRES ALEATOIRES VARIES
150 RANDOMIZE TIME
160 RG=1'RANG DU PARCOURS
170 CLS:PXM=7:PYM=INT(5*RND)+10'TIRAGE P
OSITION MAITRE
180 PXR=30:PYR=12:PPXR=PXR:PPYR=PYR'POSI
TION DEPART ROBOT ET MISE EN RESERVE POU
R TRACE
190 PEN 1:LOCATE PXR,PYR:PRINT CHR$(33):
PEN 2:LOCATE PXM,PYM:PRINT CHR$(250)'AFF
 ROBOT ET MAITRE
200 IF RG>1 THEN 1000
210 DATA 24,11,24,12,24,13,25,11,25,13,2
6,11,26,13,27,11,27,13,23,11,23,10,23,9,
24,9,23,13,23,14,23,15,24,15,22,10,21,10
,21,9,21,8,22,14,21,14,21,15,21,16,9,8,9
,10,9,12,9,14,9,16
220 DATA 8,8,8,9,8,10,8,12,8,14,8,15,8,1
6,12,8,12,9,13,9,13,10,14,10,15,10,15,14
,14,14,13,14,13,15,12,15,12,16
230 FOR N=1 TO 49:READ I,J:C(I,J)=1:PEN
1:LOCATE I,J:PRINT"=":NEXT'REPRESENTATIO
N ET AFF PILIER
240 SOUND 1,200,80,7:FOR AT=1 TO 2000:NE
XT'TOP DEPART
1000 REM BOUCLE DES MOUVEMENTS
1010 FOR M=1 TO 70'70 MOUVMTS MAXI ADMIS
 POUR ATTEINDRE MAITRE
1020 REM HEURIST NAV (1)
1030 IF PXR<>PXM THEN NUME=ABS(PYR-PYM):
DENO=ABS(PXR-PXM):IN=NUME/DENO ELSE 1060

1040 IF IN<0.5 THEN VXR=SGN(PXM-PXR):VYR
=0:GOTO 1070
1050 IF IN>2 THEN VXR=0:VYR=SGN(PYM-PYR)
:GOTO 1070
1060 VXR=SGN(PXM-PXR):VYR=SGN(PYM-PYR)
1070 TT=0'INITIALISATION DU RANG TT DES
TENTATIVES POUR TROUVER UNE POSITION LIB
RE APRES AVOIR HEURTE UN PILIER
1080 PXR=PPXR+VXR:PYR=PPYR+VYR
```

```
1090 IF PXR>29 THEN 1120 ELSE GOSUB 3000
'TEST HEURT PILIER
1100 IF OBS=0 THEN 1120 ELSE GOSUB 5000'
ESSAI NOUVELLE DIRECTION
1110 GOTO 1080'RECALCUL POSITION
1120 GOSUB 4000'TEST RETOUR EN ARRIERE
1130 IF RET=0 THEN 1150 ELSE GOSUB 5000'
ESSAI NOUV DIR
1140 GOTO 1080'RECALCUL NOUV POS
1150 PEN 3:LOCATE PPXR,PPYR:PRINT CHR$(3
3):PEN 1:LOCATE PXR,PYR:PRINT CHR$(33):S
OUND 1,30,5,7:PPXR=PXR:PPYR=PYR
1160 VVXR=VXR:VVYR=VYR'MEMO VITESE POUR
EVITER PROCHAIN RETOUR EN ARR
1170 GOSUB 6000'TEST MAITRE REJOINT
1180 NEXT M
1190 SOUND 1,500,40,7:GOTO 6030'PARCOURS
 RATE
1200 END
2000 REM ZONE DES SOUS-PROGRAMMES
3000 REM SSPROG HEURT PILIER
3010 OBS=0
3020 IF C(PXR,PYR)<>1 THEN RETURN
3030 OBS=1:TT=TT+1
3050 RETURN
4000 REM SSPROG RETOUR EN ARRIERE
4010 RET=0'NEUTRAL RET ARR PRECEDENT
4020 IF VXR<>-VVXR THEN RETURN ELSE IF V
YR<>-VVYR THEN RETURN
4030 IF TT>7 THEN RETURN ELSE :RET=1:TT=
TT+1:RETURN
5000 REM SSPROG ESSAI NOUVELLE DIRECTION
5010 REM HEURISTIQUE DE DEGAGEMENT (D):T
IRAGE 1ERE DIRECTION DE DEGAGEMENT PUIS,
EN CAS D'INSUCCES,EXPLORATION ORDONNEE D
ES AUTRES DIRECTIONS
5020 IF TT=1 THEN HAS=INT(8*RND+1):SS=IN
T(2*RND):GOTO 5060
5030 IF SS=0 THEN HAS=HAS+1::IF HAS<9 TH
EN 5060 ELSE HAS=HAS-8:GOTO 5060
5040 HAS=HAS-1:IF HAS>0 THEN 5060
5050 HAS=HAS+8
5060 IF HAS=1 THEN 5070 ELSE  IF HAS=2 T
HEN 5080 ELSE IF HAS=3 THEN 5090 ELSE IF
 HAS=4 THEN 5100 ELSE IF HAS=5 THEN 5110
 ELSE IF HAS=6 THEN 5120 ELSE IF HAS=7 T
HEN 5130 ELSE 5140
5070 VXR=1:VYR=1:GOTO 5150
5080 VXR=1:VYR=0:GOTO 5150
5090 VXR=1:VYR=-1:GOTO 5150
5100 VXR=0:VYR=-1:GOTO 5150
5110 VXR=-1:VYR=-1:GOTO 5150
5120 VXR=-1:VYR=0:GOTO 5150
5130 VXR=-1:VYR=1:GOTO 5150
```

```
5140 VXR=0:VYR=1
5150 RETURN
6000 REM SSPROG MAITRE REJOINT
6010 IF PXR<>PXM THEN RETURN ELSE IF PYR
<>PYM THEN RETURN
6020 FOR FOIS=1 TO 4:SOUND 1,27,5,7:FOR
AT=1 TO 100:NEXT:NEXT
6030 FOR AT=1 TO 800:NEXT:RG=RG+1:IF RG<
5 THEN PEN 3:LOCATE PPXR,PPYR:PRINT CHR$
(33):GOTO 180 ELSE 110
```

Pas de commentaires techniques.

ÉTAPE E12

PLUS DE COMPÉTENCE, MOINS DE LIBERTÉ

Présentation

L'heuristique de dégagement E du présent exemple se propose d'affiner l'heuristique D de E9 ; l'expérience aidant, la compétence et l'efficacité s'améliorent.

En E9, Roméo laissait au hasard le soin (quelle contradiction des mots !) de fixer la direction de la première tentative de dégagement d'un pilier, avant de poursuivre ensuite rationnellement l'exploration des autres directions en cas d'insuccès.

Mais il paraît évident, après coup, de faire partir l'exploration *du pilier lui-même*, à droite ou à gauche, car c'est ainsi qu'on obtiendra la première position libre la plus proche du but, c'est-à-dire la mieux en harmonie avec l'heuristique de navigation.

Notons qu'en fait d'évidence, il s'agit plutôt d'un théorème dont la longue démonstration passe par les déboires des étapes précédentes. Il en est de même pour toutes les évidences actuelles de ce monde. On aurait certes pu y accéder d'un seul bond, en se privant du plaisir des joyeuses escapades buissonnières de Roméo.

Pour réaliser l'heuristique E, il est nécessaire de repérer puis de mémoriser la direction d'arrivée sur le pilier, ce qui est facile puisqu'on connaît les composantes de cette vitesse d'arrivée. Le robot a simplement besoin d'un peu plus de mémoire, cette perle si méprisée des purs penseurs : sans mémoire, l'esprit est un superbe moulin broyant sa propre meule.

C'est ensuite à partir de cette direction que sera entreprise l'exploration ; le hasard conservant en pâture le sens de rotation. L'interdiction de retour en arrière subsiste, car c'est l'un des plus puissants principes d'action qu'on puisse imaginer.

Le résultat en vaut la peine, Roméo rase maintenant les piliers du labyrinthe comme un champion de slalom. Toutefois, face à un mur de piliers — ce qui se produit rarement — ce comportement provoque des retours en arrière surprenants, heureusement réprimés. On n'assiste plus à certains écarts latéraux peu compatibles avec la vénération qu'un robot doit porter à son maître.

La rançon, directement visible sur l'écran, de ce surplus d'intelligence est une réduction sensible de la variété des parcours ; le robot n'a plus, devant un pilier, que deux issues possibles, l'une immédiatement à droite,

l'autre à gauche puisque le dieu des casinos ne détient plus en son pouvoir que le sens de rotation.

Mais rien n'empêche Roméo de se relaxer de temps à autre, de renoncer à penser et de se laisser à nouveau bercer par la délicieuse heuristique B, comme font parfois les jeunes filles attendues tenant à conserver provisoirement l'illusion d'un agaçant libre arbitre.

Programme

```
100 'ETAPE E12:ROBOT CHERCHANT SON MAITR
E DANS LABYRINTHE,31 PILIERS,HEURISTIQUE
 DE NAVIGATION (1),HEURISTIQUE DE DEGAGE
MENT (E),4 PARCOURS SUR MEME TERRAIN

110 INK 0,14:INK 1,7:INK 2,20:INK 3,17:C
LS:CLEAR
120 DIM C(31,25)'TABLEAU DES COORDONNEES
 DES CASES DE L'ECRAN SUSCEPTIBLES DE RE
CEVOIR UN PILIER
130 SYMBOL AFTER 32:SYMBOL 33,&0,&7C,&D6
,&6C,&38,&FE,&7C,&6C'DESSIN ROBOT;DESSIN
 MAITRE:CODE 250
140 'NOMBRES ALEATOIRES VARIES
150 RANDOMIZE TIME
160 RG=1'RANG DU PARCOURS
170 CLS:H=INT(2*RND):IF H=1 THEN PXM=INT
(25*RND+6):PYM=6 ELSE PXM=6:PYM=INT(12*R
ND+6)'TIRAGE POSITION MAITRE
180 PXR=36:PYR=21:PPXR=PXR:PPYR=PYR'POSI
TION DEPART ROBOT ET MISE EN RESERVE POU
R TRACE
190 PEN 1:LOCATE PXR,PYR:PRINT CHR$(33):
PEN 2:LOCATE PXM,PYM:PRINT CHR$(250)'AFF
 ROBOT ET MAITRE
200 IF RG>1 THEN 1000
210 FOR N=0 TO 30:TERR$=INKEY$:IF TERR$<
>" " THEN 220 ELSE 110
220 I=INT((33-PXM)*RND)+PXM-1:J=INT((21-
PYM)/(36-PXM)*(I-PXM))+INT((5*RND-2)*(I-
PXM)/8)+PYM+1
230 IF J>PYM-2 THEN 250
240 J=PYM-1:GOTO 270
250 IF J<>PYM THEN IF J<22 THEN 270 ELSE
 J=21:GOTO 270
260 IF I<>PXM THEN 270 ELSE J=PYM-1
270 PEN 1:LOCATE I,J:PRINT"=":C(I,J)=1:N
EXT'AFF ET REPRESENTATION PILIER
280 SOUND 1,200,80,7:FOR AT=1 TO 2000:NE
XT'TOP DEPART
1000 REM BOUCLE DES MOUVEMENTS
```

```
1010 FOR M=1 TO 50'50 MOUVMTS MAXI ADMIS
 POUR ATTEINDRE MAITRE
1020 REM HEURIST NAV (1)
1030 IF PXR<>PXM THEN NUME=ABS(PYR-PYM):
DENO=ABS(PXR-PXM):IN=NUME/DENO ELSE 1060

1040 IF IN<0.5 THEN VXR=SGN(PXM-PXR):VYR
=0:GOTO 1070
1050 IF IN>2 THEN VXR=0:VYR=SGN(PYM-PYR)
:GOTO 1070
1060 VXR=SGN(PXM-PXR):VYR=SGN(PYM-PYR)
1070 TT=0'INITIALISATION DU RANG TT DES
TENTATIVES POUR TROUVER UNE POSITION LIB
RE APRES AVOIR HEURTE UN PILIER
1080 PXR=PPXR+VXR:PYR=PPYR+VYR
1090 IF PXR>31 THEN 1120 ELSE GOSUB 3000
'TEST HEURT PILIER
1100 IF OBS=0 THEN 1110 ELSE 1080'RECALC
UL POS
1110 IF RET=0 THEN 1120 ELSE 1080
1120 PEN 3:LOCATE PPXR,PPYR:PRINT CHR$(3
3):PEN 1:LOCATE PXR,PYR:PRINT CHR$(33):S
OUND 1,30,5,7:PPXR=PXR:PPYR=PYR
1130 VVXR=VXR:VVYR=VYR'MEMO VITESE POUR
EVITER PROCHAIN RETOUR EN ARR
1140 GOSUB 4000'TEST MAITRE REJOINT
1150 FOR N=1 TO 80:NEXT:NEXT M
1160 SOUND 1,500,40,7:GOTO 4030'PARCOURS
 RATE
1170 END
2000 REM ZONE DES SOUS-PROGRAMMES
3000 REM SSPROG HEURT PILIER ET/OU RETOU
R EN ARRIERE
3010 OBS=0:RET=0
3020 REM TEST PILIER
3030 IF C(PXR,PYR)=1 THEN 3040 ELSE 3050
3040 FOR AT=1 TO 100:NEXT:SOUND 1,50,5,7
:PEN 2:LOCATE PXR,PYR:PRINT"*":FOR AT=1
TO 50:NEXT:PEN 1:LOCATE PXR,PYR:PRINT"="
:OBS=1:GOTO 3080
3050 REM TEST RETOUR ARRIERE
3060 IF VXR<>-VVXR THEN RETURN ELSE IF V
YR<>-VVYR THEN RETURN
3070 IF TT>7 THEN RETURN ELSE FOR AT=1 T
O 100:NEXT AT:SOUND 1,20,5,7:PEN 2:LOCAT
E PXR,PYR:PRINT"*":FOR AT=1 TO 150:NEXT:
PEN 3:LOCATE PXR,PYR:PRINT CHR$(33):RET=
1
3080 TT=TT+1
3090 REM ESSAI NOUVELLE DIRECTION
3100 REM HEURISTIUQE DE DEGAGEMENT (E):E
XPLORATION ORDONNEE DES AUTRES DIRECIONS
 EN PARTANT DU PILIER HEURTE
3110 FOR AT=1 TO 100:NEXT AT:IF TT>1 THE
```

```
N 3250'CAS 2EME PASSAGE ET SUIVANTS
3120 IF VXR=-1 THEN 3150'CAS 1ER PASSAGE
,DETERMINATION DE LA DIRECTION D1 DU 1ER
 HEURT
3130 IF VXR=0 THEN 3180
3140 IF VXR =1 THEN 3200
3150 IF VYR=-1 THEN D1=5:GOTO 3230
3160 IF VYR=0 THEN D1=6:GOTO 3230
3170 IF VYR=1 THEN D1=7:GOTO 3230
3180 IF VYR=-1 THEN D1=4:GOTO 3230
3190 IF VYR=1 THEN D1=8:GOTO 3230
3200 IF VYR=-1 THEN D1=3:GOTO 3230
3210 IF VYR=0 THEN D1=2:GOTO 3230
3220 IF VYR=1 THEN D1=1
3230 D=D1'DIRECTION D DEBUT DE L'EXPLORA
TION
3240 SS=INT(2*RND)'TIRAGE SENS DE L'EXPL
ORATION
3250 REM EXPLORATION POSITION SUIVANTE
3260 IF SS=0 THEN D=D+1:IF D<9 THEN 3280
 ELSE D=D-8:GOTO 3280
3270 D=D-1:IF D>0 THEN 3280 ELSE D=D+8
3280 IF D=1 THEN 3290 ELSE IF D=2 THEN 3
300 ELSE IF D=3 THEN 3310 ELSE IF D=4 TH
EN 3320 ELSE IF D=5 THEN 3330 ELSE IF D=
6 THEN 3340 ELSE IF D=7 THEN 3350 ELSE 3
360
3290 VXR=1:VYR=1:GOTO 3370
3300 VXR=1:VYR=0:GOTO 3370
3310 VXR=1:VYR=-1:GOTO 3370
3320 VXR=0:VYR=-1:GOTO 3370
3330 VXR=-1:VYR=-1:GOTO 3370
3340 VXR=-1:VYR=0:GOTO 3370
3350 VXR=-1:VYR=1:GOTO 3370
3360 VXR=0:VYR=1
3370 RETURN
4000 REM SSPROG MAITRE REJOINT
4010 IF PXR<>PXM THEN RETURN ELSE IF PYR
<>PYM THEN RETURN
4020 FOR FOIS=1 TO 4:SOUND 1,27,5,7:FOR
AT=1 TO 100:NEXT:NEXT
4030 FOR AT=1 TO 800:NEXT:RG=RG+1:IF RG<
5 THEN PEN 3:LOCATE PPXR,PPYR:PRINT CHR$
(33):GOTO 180 ELSE 110
```

Commentaires techniques

- Les trois sous-programmes de E9 (heurt pilier, retour en arrière et essai nouvelle direction) ont été réunis en un seul.
- Après détection d'un pilier au début de ce sous-programme, le repérage de la direction d'arrivée se fait à l'aide du paramètre D1 dont

la valeur (de 1 à 8) est déterminée en fonction des valeurs VXR et VYR de la vitesse d'arrivée, suivant la même correspondance qu'en E8. On obtient ainsi huit directions possibles :

VXR = 1 et VYR = 1, D1 = 1	VXR = −1 et VYR = −1, D1 = 5
VXR = 1 et VYR = 0, D1 = 2	VXR = −1 et VYR = 0, D1 = 6
VXR = 1 et VYR = −1, D1 = 3	VXR = −1 et VYR = 1, D1 = 7
VXR = 0 et VYR = −1, D1 = 4	VXR = 0 et VYR = 1, D1 = 8

Dans l'ordre, ces valeurs de D1 correspondent respectivement au sud-est, à l'est, au nord-est, au nord, au nord-ouest, à l'ouest, au sud-ouest et au sud. C'est cette valeur D1 qui remplace le paramètre HAS tiré au sort de E9.

- Le sens d'exploration SS reste tiré au hasard comme dans E9.

ÉTAPE E13

LES FONCTIONS D'ÉVALUATION

Présentation

Gravissons l'ultime marche et examinons ce que peut être une certaine heuristique de dégagement F meilleure que la précédente. Sans doute plusieurs lecteurs l'auront-ils déjà débusquée.

Nous accéderons plus loin, en E27 et E28, à d'autres heuristiques ressortissant à un concept différent.

Mais, dans son mode de pensée actuel, au premier degré de ses perceptions sensorielles, Roméo arrive ici au bout de ses possibilités. Toute fantaisie va lui être refusée [3], sa pensée envahissante agissant maintenant en dictatrice.

Au déterminisme bêtifiant de la première heuristique de dégagement A va se substituer un autre déterminisme triomphant dont le noyautage inexorable s'est opéré sous nos yeux au cours des cinq promenades, de plus en plus sages, des programmes précédents. L'on choisira parmi ces modes selon son cœur ou selon sa tête, un peu comme entre Vercingétorix et César.

Au présent stade donc, le robot retire au hasard son dernier droit, celui de choisir le sens d'exploration de l'heuristique E, et estime qu'il est assez grand pour *effectuer les deux explorations puis choisir la meilleure des deux positions ainsi obtenues.* Il commence donc par simuler les deux explorations, inscrit un point d'interrogation sur chacune des deux positions ainsi repérées, *évalue* ces deux positions et adopte la meilleure.

L'outil utilisé pour soupeser la valeur des positions s'appelle une *fonction d'évaluation*. La programmation du jeu d'échecs, par exemple, a donné lieu à l'invention d'une foule de fonctions d'évaluation, la nôtre possède une double simplicité :

- l'évaluation se fait sur le pas en cours, *à la profondeur zéro*, sans s'occuper des aléas ultérieurs éventuels (piliers) ;
- l'évaluation consiste simplement à apprécier la distance de la position au maître, et à choisir la plus courte ou, en cas d'égalité, la dernière calculée.

3. À un détail près : si deux solutions s'avèrent équivalentes, il adoptera la dernière examinée.

Mais il est essentiel de noter que cette ''distance'' robot-maître est définie d'une manière spéciale, adaptée au problème à résoudre et qui ne correspond pas forcément à la définition mathématique classique. C'est en quelque sorte une *distance heuristique*, analogue à celle que peut apprécier d'un coup d'œil le joueur de pétanque en tenant compte des accidents du terrain.

Dans le cas présent, on a choisi la définition la plus naturelle : le nombre de pas qui séparent le robot de son maître, en supposant qu'il n'y ait plus ensuite de piliers sur le trajet.

Il est facile de voir, sur la figure de l'étape E4, qu'il s'agit, pour une position donnée du robot, de la plus grande des deux distances décomptées verticalement ou horizontalement. Par exemple, sur la figure en question, la distance horizontale est 17, la distance verticale est 5, il faut donc que le robot fasse dans le meilleur des cas 17 pas (horizontaux ou en biais) pour rejoindre son maître.

Nous avions d'ailleurs remarqué que ce nombre de pas nécessaires est identique pour plusieurs heuristiques de navigation possibles et qu'il constitue bien un nombre caractéristique de la position. Le robot est donc censé posséder une vue qui lui permette d'apprécier ce genre de distance et un intellect pour procéder aux comparaisons.

On aurait certes pu choisir, pour évaluer la position, la distance au sens commun du terme, c'est-à-dire la longueur de la droite joignant la position au maître, calculable à l'aide du théorème de Pythagore, mais cette manière, outre son formalisme et sa lourdeur, n'eût pas été adaptée au problème posé.

Nous verrons en E30 une fonction d'évaluation des distances différemment exprimée, tenant compte des particularités d'une autre situation.

Ces façons de procéder sont parfaitement logiques puisqu'en fait on a à évaluer des *situations globales*, avec leurs aspects quantitatifs et qualitatifs parfois complexes, très difficiles à exprimer.

Une fonction d'évaluation peut ainsi contenir de nombreux paramètres, à l'intérieur de structures plus ou moins bizarres, linéaires ou non, affectés de coefficients empiriques rationnellement inexplicables, et dont le programmeur peut faire varier le dosage en fonction... de son talent.

Nous verrons l'une de ces formules magiques au sein du programme E27. Elles élèvent en quelque sorte les mathématiques au vrai niveau que leur conférait Évariste Galois, celui des idées, bien au-delà des calculs.

L'efficacité de l'heuristique F apparaît à l'écran, mais d'une manière relativement discrète par rapport à l'heuristique E. Il faut peut-être y voir l'effet logarithmique de la relation cause-effet, déjà signalé en E5 à propos des heuristiques de navigation 1 et 2, ou plutôt le caractère apparemment anodin et naturel des actes mûrement réfléchis.

Programme

```
100 'ETAPE E13:ROBOT CHERCHANT SON MAITR
E DANS LABYRINTHE,31 PILIERS,HEURISTIQUE
 DE NAVIGATION (1),HEURISTIQUE DE DEGAGE
MENT (F),4 PARCOURS SUR MEME TERRAIN

110 INK 0,22:INK 1,1:INK 2,6:INK 3,14:CL
S:CLEAR
120 DIM C(32,25)'TABLEAU DES COORDONNEES
 DES CASES DE L'ECRAN SUSCEPTIBLES DE RE
CEVOIR UN PILIER
130 SYMBOL AFTER 32:SYMBOL 33,&0,&7C,&D6
,&6C,&38,&FE,&7C,&6C'DESSIN ROBOT;DESSIN
 MAITRE:CODE 250
140 'NOMBRES ALEATOIRES VARIES
150 RANDOMIZE TIME
160 RG=1'RANG DU PARCOURS
170 CLS:H=INT(2*RND):IF H=1 THEN PXM=INT
(25*RND+6):PYM=6 ELSE PXM=6:PYM=INT(12*R
ND+6)'TIRAGE POSITION MAITRE
180 PXR=36:PYR=21:PPXR=PXR:PPYR=PYR'POSI
TION DEPART ROBOT ET MISE EN RESERVE POU
R TRACE
190 PEN 1:LOCATE PXR,PYR:PRINT CHR$(33):
PEN 2:LOCATE PXM,PYM:PRINT CHR$(250)'AFF
 ROBOT ET MAITRE
200 IF RG>1 THEN 1000
210 FOR N=0 TO 30:TERR$=INKEY$:IF TERR$<
>" " THEN 220 ELSE 110
220 I=INT((33-PXM)*RND)+PXM-1:J=INT((21-
PYM)/(36-PXM)*(I-PXM))+INT((5*RND-2)*(I-
PXM)/8)+PYM+1
230 IF J>PYM-2 THEN 250
240 J=PYM-1:GOTO 270
250 IF J<>PYM THEN IF J<22 THEN 270 ELSE
 J=21:GOTO 270
260 IF I<>PXM THEN 270 ELSE J=PYM-1
270 PEN 1:LOCATE I,J:PRINT"=":C(I,J)=1:N
EXT'AFF ET REPRESENTATION PILIER
280 SOUND 1,200,80,7:FOR AT=1 TO 2000:NE
XT'TOP DEPART
1000 REM BOUCLE DES MOUVEMENTS
1010 FOR M=1 TO 50'50 MOUVMTS MAXI ADMIS
 POUR ATTEINDRE MAITRE
1020 REM HEURIST NAV (1)
1030 IF PXR<>PXM THEN NUME=ABS(PYR-PYM):
DENO=ABS(PXR-PXM):IN=NUME/DENO ELSE 1060

1040 IF IN<0.5 THEN VXR=SGN(PXM-PXR):VYR
=0:GOTO 1070
1050 IF IN>2 THEN VXR=0:VYR=SGN(PYM-PYR)
:GOTO 1070
1060 VXR=SGN(PXM-PXR):VYR=SGN(PYM-PYR)
```

```
1070 TT=0'INITIALISATION DU RANG TT DES
TENTATIVES POUR TROUVER UNE POSITION LIB
RE APRES AVOIR HEURTE UN PILIER
1080 PXR=PPXR+VXR:PYR=PPYR+VYR
1090 IF PXR>32 THEN 1260 ELSE GOSUB 3000
'TEST HEURT PILIER
1100 IF OBS=0 THEN 1110 ELSE 1080'RECALC
UL POS
1110 IF RET=0 THEN 1120 ELSE 1080
1120 IF M<>MPI THEN 1260
1130 REM HEURIST DE DEGT (F):ESSAI AUTRE
 SENS D'EXPLORATION
1140 IF ESS=1 THEN 1180 ELSE PEN 2:LOCAT
E PXR,PYR:PRINT"?"::PXR1=PXR:PYR1=PYR:VX
R1=VXR:VYR1=VYR'AFF ET MEMO 1ERE POS
1150 FOR AT=1 TO 100:NEXT:SSS=SS:IF SSS=
1 THEN SS=0 ELSE SS=1'INVERSION SENS DE
ROTATION
1160 TT=2:ESS=1:D=D1:GOSUB 3250
1170 GOTO 1080'RECALCUL POS
1180 ESS=0:MPI=0'REINIT ESS ET MPI
1190 PEN 2:LOCATE PXR,PYR:PRINT"?"'AFF 2
EME POS
1200 FOR AT=1 TO 300: NEXT
1210 REM HEUR DE DEGT (F):CHOIX MEILLEUR
E POSITION,PLUS COURTE DES 2 DISTANCES D
IST1 ET DIST2 AU MAITRE
1220 HOR1=ABS(PXM-PXR1):VERT1=ABS(PYM-PY
R1)::IF HOR1>VERT1 THEN DIST1=HOR1 ELSE
DIST1=VERT1
1230 HOR2=ABS(PXM-PXR):VERT2=ABS(PYM-PYR
):IF HOR2>VERT2 THEN DIST2=HOR2 ELSE DIS
T2=VERT2
1240 IF DIST2<=DIST1 THEN 1250 ELSE LOCA
TE PXR,PYR:PRINT" ":PXR=PXR1:PYR=PYR1:VX
R=VXR1:VYR=VYR1:GOTO 1260'1ERE POS CHOIS
IE
1250 LOCATE PXR1,PYR1:PRINT" "'2EME POS
CHOISIE
1260 PEN 3:LOCATE PPXR,PPYR:PRINT CHR$(3
3):PEN 1:LOCATE PXR,PYR:PRINT CHR$(33):S
OUND 1,30,5,7:PPXR=PXR:PPYR=PYR:VVXR=VXR
:VVYR=VYR
1270 GOSUB 4000'TEST MAITRE REJOINT
1280 FOR N=1 TO 80:NEXT:NEXT M
1290 SOUND 1,500,40,7:GOTO 4030'PARCOURS
 RATE
1300 END
2000 REM ZONE DES SOUS-PROGRAMMES
3000 REM SSPROG HEURT PILIER ET/OU RETOU
R EN ARRIERE
3010 OBS=0:RET=0
3020 REM TEST PILIER
3030 IF C(PXR,PYR)=1 THEN 3040 ELSE 3050
```

```
3040 FOR AT=1 TO 100:NEXT:SOUND 1,50,5,7
:PEN 2:LOCATE PXR,PYR:PRINT"*":FOR AT=1
TO 50:NEXT:PEN 1:LOCATE PXR,PYR:PRINT"="
:OBS=1:GOTO 3080
3050 REM TEST RETOUR ARRIERE
3060 IF VXR<>-VVXR THEN RETURN ELSE IF V
YR<>-VVYR THEN RETURN
3070 IF TT>7 THEN RETURN ELSE FOR AT=1 T
O 100:NEXT AT:SOUND 1,20,5,7:PEN 2:LOCAT
E PXR,PYR:PRINT"*":FOR AT=1 TO 150:NEXT:
PEN 3:LOCATE PXR,PYR:PRINT CHR$(33):RET=
1
3080 FOR AT=1 TO 150:NEXT:TT=TT+1:MPI=M'
MEMO NO BOUCLE
3090 REM ESSAI NOUVELLE DIRECTION
3100 REM HEURISTIQUE DE DEGAGEMENT (F):E
XPLORATION ORDONNEE DES  DIRECTIONS EN P
ARTANT DU PILIER HEURTE
3110 FOR AT=1 TO 100:NEXT AT:IF TT>1 THE
N 3250'CAS 2EME PASSAGE ET SUIVANTS
3120 IF VXR=-1 THEN 3150'CAS 1ER PASSAGE
,DETERMINATION DE LA DIRECTION D1 DU 1ER
 HEURT
3130 IF VXR=0 THEN 3180
3140 IF VXR =1 THEN 3200
3150 IF VYR=-1 THEN D1=5:GOTO 3230
3160 IF VYR=0 THEN D1=6:GOTO 3230
3170 IF VYR=1 THEN D1=7:GOTO 3230
3180 IF VYR=-1 THEN D1=4:GOTO 3230
3190 IF VYR=1 THEN D1=8:GOTO 3230
3200 IF VYR=-1 THEN D1=3:GOTO 3230
3210 IF VYR=0 THEN D1=2:GOTO 3230
3220 IF VYR=1 THEN D1=1
3230 D=D1'DIRECTION D DEBUT DE L'EXPLORA
TION
3240 SS=INT(2*RND)'TIRAGE SENS DE L'EXPL
ORATION
3250 REM EXPLORATION POSITION SUIVANTE
3260 IF SS=0 THEN D=D+1:IF D<9 THEN 3280
 ELSE D=D-8:GOTO 3280
3270 D=D-1:IF D>0 THEN 3280 ELSE D=D+8
3280 IF D=1 THEN 3290 ELSE IF D=2 THEN 3
300 ELSE IF D=3 THEN 3310 ELSE IF D=4 TH
EN 3320 ELSE IF D=5 THEN 3330 ELSE IF D=
6 THEN 3340 ELSE IF D=7 THEN 3350 ELSE 3
360
3290 VXR=1:VYR=1:GOTO 3370
3300 VXR=1:VYR=0:GOTO 3370
3310 VXR=1:VYR=-1:GOTO 3370
3320 VXR=0:VYR=-1:GOTO 3370
3330 VXR=-1:VYR=-1:GOTO 3370
3340 VXR=-1:VYR=0:GOTO 3370
3350 VXR=-1:VYR=1:GOTO 3370
3360 VXR=0:VYR=1
```

```
3370 RETURN
4000 REM SSPROG MAITRE REJOINT
4010 IF PXR<>PXM THEN RETURN ELSE IF PYR
<>PYM THEN RETURN
4020 FOR FOIS=1 TO 4:SOUND 1,27,5,7:FOR
AT=1 TO 100:NEXT:NEXT
4030 FOR AT=1 TO 800:NEXT:RG=RG+1:IF RG<
5 THEN PEN 3:LOCATE PPXR,PPYR:PRINT CHR$
(33):GOTO 180 ELSE 110
```

Commentaires techniques

- Au passage dans le sous-programme "heurt pilier ou retour en arrière", chaque fois qu'un empêchement (pilier ou retour en arrière) entrave la marche du robot, le numéro de la boucle en cours (MPI) est mémorisé et une nouvelle direction est tentée dans un certain sens d'exploration SS tiré au sort, jusqu'à trouver une position libre.
- Dans une telle boucle (pour laquelle M = MPI), le programme provoque alors une seconde exploration dans le sens inverse. Les deux explorations sont conduites sous l'égide du paramètre ESS qui vaut 0 pour le premier sens puis 1 pour le second. Les deux positions possibles sont mémorisées et visualisées au moyen d'un point d'interrogation.
- Le programme calcule les deux distances DIST1 et DIST2, chacune étant la plus élevée des distances horizontale et verticale HOR1 et VERT1, puis HOR2 et VERT2.
- Les valeurs DIST1 et DIST2 sont alors comparées, et la plus courte est choisie.
- Chaque fois, la vitesse du robot est mémorisée (VVXR, VVYR) en vue d'interdire un prochain retour en arrière éventuel.
- Quatre parcours successifs sont effectués sur le même terrain et permettent de déceler encore un léger indéterminisme ; en effet, si les distances DIST1 et DIST2 sont égales, c'est la seconde position explorée qui est adoptée, et cela peut varier d'un parcours à l'autre puisque le premier sens de rotation est tiré au hasard.

ÉTAPE E14

SILENCE, ON TOURNE

Présentation

Il s'agit du même programme que E13, mais avec suppression du dispositif audiovisuel permettant de suivre les opérations mentales du robot, comme E11 vis-à-vis de E10.

Il n'est pas facile, pour le spectateur non averti, de deviner alors quelle est la tactique du robot. Il peut seulement en remarquer l'efficacité, la sûreté, mises en relief par le léger temps de réflexion qui marque chaque contournement de pilier.

Au plan philosophique, on peut se demander pour E13 et E14 ce qui autorise à croire en l'efficacité de cette heuristique F. En effet, elle ne tient aucun compte de la rencontre ultérieure possible (et même probable) d'autres piliers, peut-être plus nombreux du côté de la position choisie que du côté de la position dédaignée.

La réponse est claire : comme Roméo procède ainsi devant chaque pilier rencontré, la méthode est *en moyenne* meilleure que celle qui consisterait à prendre la première position venue. Le lecteur pourra s'en convaincre concrètement en mettant dans un même programme les deux heuristiques E et F et en comparant les nombres de pas moyens nécessaires à l'exécution d'un même parcours.

Bien entendu, pour faire mieux, il faudrait voir un peu plus loin... mais n'anticipons pas.

Programme

```
100 'ETAPE E14:COMME E13 SANS SONO NI VI
SUALISATION DES HEURISTIQUES
110 INK 0,22:INK 1,1:INK 2,6:INK 3,14:CL
S:CLEAR
120 DIM C(32,25)'TABLEAU DES COORDONNEES
 DES CASES DE L'ECRAN SUSCEPTIBLES DE RE
CEVOIR UN PILIER
130 SYMBOL AFTER 32:SYMBOL 33,&0,&7C,&D6
,&6C,&38,&FE,&7C,&6C'DESSIN ROBOT;DESSIN
 MAITRE:CODE 250
140 'NOMBRES ALEATOIRES VARIES
150 RANDOMIZE TIME
160 RG=1'RANG DU PARCOURS
```

```
170 CLS:H=INT(2*RND):IF H=1 THEN PXM=INT
(25*RND+6):PYM=6 ELSE PXM=6:PYM=INT(12*R
ND+6)'TIRAGE POSITION MAITRE
180 PXR=36:PYR=21:PPXR=PXR:PPYR=PYR'POSI
TION DEPART ROBOT ET MISE EN RESERVE POU
R TRACE
190 PEN 1:LOCATE PXR,PYR:PRINT CHR$(33):
PEN 2:LOCATE PXM,PYM:PRINT CHR$(250)'AFF
 ROBOT ET MAITRE
200 IF RG>1 THEN 1000
210 FOR N=0 TO 30:TERR$=INKEY$:IF TERR$<
>" " THEN 220 ELSE 110
220 I=INT((33-PXM)*RND)+PXM-1:J=INT((21-
PYM)/(36-PXM)*(I-PXM))+INT((5*RND-2)*(I-
PXM)/8)+PYM+1
230 IF J>PYM-2 THEN 250
240 J=PYM-1:GOTO 270
250 IF J<>PYM THEN IF J<22 THEN 270 ELSE
 J=21:GOTO 270
260 IF I<>PXM THEN 270 ELSE J=PYM-1
270 PEN 1:LOCATE I,J:PRINT"=":C(I,J)=1:N
EXT'AFF ET REPRESENTATION PILIER
280 SOUND 1,200,80,7:FOR AT=1 TO 2000:NE
XT'TOP DEPART
1000 REM BOUCLE DES MOUVEMENTS
1010 FOR M=1 TO 50'50 MOUVMTS MAXI ADMIS
 POUR ATTEINDRE MAITRE
1020 REM HEURIST NAV (1)
1030 IF PXR<>PXM THEN NUME=ABS(PYR-PYM):
DENO=ABS(PXR-PXM):IN=NUME/DENO ELSE 1060

1040 IF IN<0.5 THEN VXR=SGN(PXM-PXR):VYR
=0:GOTO 1070
1050 IF IN>2 THEN VXR=0:VYR=SGN(PYM-PYR)
:GOTO 1070
1060 VXR=SGN(PXM-PXR):VYR=SGN(PYM-PYR)
1070 TT=0'INITIALISATION DU RANG TT DES
TENTATIVES POUR TROUVER UNE POSITION LIB
RE APRES AVOIR HEURTE UN PILIER
1080 PXR=PPXR+VXR:PYR=PPYR+VYR
1090 IF PXR>32 THEN 1240 ELSE GOSUB 3000
'TEST HEURT PILIER
1100 IF OBS=0 THEN 1110 ELSE 1080'RECALC
UL POS
1110 IF RET=0 THEN 1120 ELSE 1080
1120 IF M<>MPI THEN 1240
1130 REM HEURIST DE DEGT (F):ESSAI AUTRE
 SENS D'EXPLORATION
1140 IF ESS=1 THEN 1180 ELSE PXR1=PXR:PY
R1=PYR:VXR1=VXR:VYR1=VYR'AFF ET MEMO 1ER
E POS
1150 SSS=SS:IF SSS=1 THEN SS=0 ELSE SS=1
'INVERSION SENS DE ROTATION
1160 TT=2:ESS=1:D=D1:GOSUB 3250
```

```
1170 GOTO 1080'RECALCUL POS
1180 ESS=0:MPI=0'REINIT ESS ET MPI
1190 REM HEUR DE DEGT (F):CHOIX MEILLEUR
E POSITION,PLUS COURTE DES 2 DISTANCES D
IST1 ET DIST2 AU MAITRE
1200 HOR1=ABS(PXM-PXR1):VERT1=ABS(PYM-PY
R1)::IF HOR1>VERT1 THEN DIST1=HOR1 ELSE
DIST1=VERT1
1210 HOR2=ABS(PXM-PXR):VERT2=ABS(PYM-PYR
):IF HOR2>VERT2 THEN DIST2=HOR2 ELSE DIS
T2=VERT2
1220 IF DIST2<=DIST1 THEN 1230 ELSE PXR=
PXR1:PYR=PYR1:VXR=VXR1:VYR=VYR1:GOTO 124
0'1ERE POS CHOISIE
1230 VVXR=VXR:VVYR=VYR'2EME POS CHOISIE
1240 PEN 3:LOCATE PPXR,PPYR:PRINT CHR$(3
3):PEN 1:LOCATE PXR,PYR:PRINT CHR$(33):S
OUND 1,30,5,7:PPXR=PXR:PPYR=PYR
1250 GOSUB 4000'TEST MAITRE REJOINT
1260 NEXT M
1270 SOUND 1,500,40,7:GOTO 4030'PARCOURS
 RATE
1280 END
2000 REM ZONE DES SOUS-PROGRAMMES
3000 REM SSPROG HEURT PILIER ET/OU RETOU
R EN ARRIERE
3010 OBS=0:RET=0
3020 REM TEST PILIER
3030 IF C(PXR,PYR)=1 THEN 3040 ELSE 3050
3040 OBS=1:GOTO 3080
3050 REM TEST RETOUR ARRIERE
3060 IF VXR<>-VVXR THEN RETURN ELSE IF V
YR<>-VVYR THEN RETURN
3070 IF TT>7 THEN RETURN ELSE RET=1
3080 TT=TT+1:MPI=M'MEMO NO BOUCLE
3090 REM ESSAI NOUVELLE DIRECTION
3100 REM HEURISTIQUE DE DEGAGEMENT (F):E
XPLORATION ORDONNEE DES  DIRECTIONS EN P
ARTANT DU PILIER HEURTE
3110 IF TT>1 THEN 3250'CAS 2EME PASSAGE
ET SUIVANTS
3120 IF VXR=-1 THEN 3150'CAS 1ER PASSAGE
,DETERMINATION DE LA DIRECTION D1 DU 1ER
 HEURT
3130 IF VXR=0 THEN 3180
3140 IF VXR =1 THEN 3200
3150 IF VYR=-1 THEN D1=5:GOTO 3230
3160 IF VYR=0 THEN D1=6:GOTO 3230
3170 IF VYR=1 THEN D1=7:GOTO 3230
3180 IF VYR=-1 THEN D1=4:GOTO 3230
3190 IF VYR=1 THEN D1=8:GOTO 3230
3200 IF VYR=-1 THEN D1=3:GOTO 3230
3210 IF VYR=0 THEN D1=2:GOTO 3230
3220 IF VYR=1 THEN D1=1
```

```
3230 D=D1'DIRECTION D DEBUT DE L'EXPLORA
TION
3240 SS=INT(2*RND)'TIRAGE SENS DE L'EXPL
ORATION
3250 REM EXPLORATION POSITION SUIVANTE
3260 IF SS=0 THEN D=D+1:IF D<9 THEN 3280
 ELSE D=D-8:GOTO 3280
3270 D=D-1:IF D>0 THEN 3280 ELSE D=D+8
3280 IF D=1 THEN 3290 ELSE IF D=2 THEN 3
300 ELSE IF D=3 THEN 3310 ELSE IF D=4 TH
EN 3320 ELSE IF D=5 THEN 3330 ELSE IF D=
6 THEN 3340 ELSE IF D=7 THEN 3350 ELSE 3
360
3290 VXR=1:VYR=1:GOTO 3370
3300 VXR=1:VYR=0:GOTO 3370
3310 VXR=1:VYR=-1:GOTO 3370
3320 VXR=0:VYR=-1:GOTO 3370
3330 VXR=-1:VYR=-1:GOTO 3370
3340 VXR=-1:VYR=0:GOTO 3370
3350 VXR=-1:VYR=1:GOTO 3370
3360 VXR=0:VYR=1
3370 RETURN
4000 REM SSPROG MAITRE REJOINT
4010 IF PXR<>PXM THEN RETURN ELSE IF PYR
<>PYM THEN RETURN
4020 FOR FOIS=1 TO 4:SOUND 1,27,5,7:FOR
AT=1. TO 100:NEXT:NEXT
4030 FOR AT=1 TO 800:NEXT:RG=RG+1:IF RG<
5 THEN PEN 3:LOCATE PPXR,PPYR:PRINT CHR$
(33):GOTO 180 ELSE 110
```

Pas de commentaires techniques.

ÉTAPE E15

DE LA RIGUEUR DANS LA TACTIQUE, DU FLOU[4] DANS LA STRATÉGIE

Présentation

L'abandon, en E13-E14, de toute fantaisie au profit de l'efficacité est peu conforme au caractère de Roméo qui tient beaucoup à conserver un certain libre arbitre et à se promener en dehors des sentiers battus du labyrinthe.

C'est sous l'aiguillon de cette regrettable indiscipline qu'il a alors une idée véritablement géniale. Il a constaté, de E7 à E13, la corrélation étroite entre la qualité des *heuristiques de dégagement* et celle de ses prestations, donc rien à redire de ce côté.

Mais il s'est souvenu aussi que la *navigation générale* hors piliers pouvait supporter sans inconvénient (nombre de pas nécessaires) quelques accommodements, tel le passage de l'heuristique 1 à l'heuristique 2.

Il a en conséquence décidé de s'offrir une petite heuristique de navigation spécialement adaptée, baptisée 3, et qui est une heuristique 1 "améliorée". L'amélioration consiste à y introduire de la fantaisie à chaque pas à l'aide du générateur de hasard. Ce qui prouve que, si le soldat doit rester pur, le général peut se permettre quelques écarts.

Ce résultat est obtenu en remplaçant les valeurs habituelles des composantes VXR et VYR de l'heuristique 1 par des valeurs floues qui oscillent aléatoirement autour des précédentes.

Il y a une infinité de manières de doser ce flou. On peut le mettre dans le paramètre IN défini en E4 (inclinaison de la direction robot-maître) en écrivant par exemple :

$$IN = NUME/DENO + INT(3 \times RND - 1)$$

auquel cas IN prendra soit sa vraie valeur NUME/DENO, soit cette valeur augmentée ou diminuée de 1.

On peut aussi, comme proposé dans le programme démonstratif ci-après, introduire la zizanie directement dans les composantes de la vitesse. Deux conséquences, l'une évidente, l'autre étrange :

4. Ce flou n'a rien à voir avec la théorie des ensembles flous de L. Zadeh.

- Roméo redevient soudainement joyeux et semble avoir acquis un sens supplémentaire, le goût des bons vins ;
- Il arrive aussi sûrement à destination que s'il était à jeun, et presque aussi rapidement. Les piliers semblent le ramener à la raison, ce qui est normal puisque l'heuristique de dégagement F reste intacte.

Il lui est arrivé malgré tout une fois ou deux de sortir de l'écran à sa partie inférieure, terrassé par le flou.

Programme

```
100 'ETAPE E15:COMME E14,AVEC HEURISTIQUE
 (3):INTRODUCTION D'UNE CERTAINE MARGE D
E HASARD DANS L'HEURISTIQUE (1)
110 INK 0,22:INK 1,1:INK 2,6:INK 3,14:CL
S:CLEAR
120 DIM C(32,25)'TABLEAU DES COORDONNEES
 DES CASES DE L'ECRAN SUSCEPTIBLES DE RE
CEVOIR UN PILIER
130 SYMBOL AFTER 32:SYMBOL 33,&0,&7C,&D6
,&6C,&38,&FE,&7C,&6C'DESSIN ROBOT;DESSIN
 MAITRE:CODE 250
140 'NOMBRES ALEATOIRES VARIES
150 RANDOMIZE TIME
160 RG=1'RANG DU PARCOURS
170 CLS:H=INT(2*RND):IF H=1 THEN PXM=INT
(25*RND+6):PYM=6 ELSE PXM=6:PYM=INT(12*R
ND+6)'TIRAGE POSITION MAITRE
180 PXR=36:PYR=21:PPXR=PXR:PPYR=PYR'POSI
TION DEPART ROBOT ET MISE EN RESERVE POU
R TRACE
190 PEN 1:LOCATE PXR,PYR:PRINT CHR$(33):
PEN 2:LOCATE PXM,PYM:PRINT CHR$(250)'AFF
 ROBOT ET MAITRE
200 IF RG>1 THEN 1000
210 FOR N=0 TO 30:TERR$=INKEY$:IF TERR$<
>" " THEN 220 ELSE 110
220 I=INT((33-PXM)*RND)+PXM-1:J=INT((21-
PYM)/(36-PXM)*(I-PXM))+INT((5*RND-2)*(I-
PXM)/8)+PYM+1
230 IF J>PYM-2 THEN 250
240 J=PYM-1:GOTO 270
250 IF J<>PYM THEN IF J<22 THEN 270 ELSE
 J=21:GOTO 270
260 IF I<>PXM THEN 270 ELSE J=PYM-1
270 PEN 1:LOCATE I,J:PRINT"=":C(I,J)=1:N
EXT'AFF ET REPRESENTATION PILIER
280 SOUND 1,200,80,7:FOR AT=1 TO 2000:NE
XT'TOP DEPART
```

```
1000 REM BOUCLE DES MOUVEMENTS
1010 FOR M=1 TO 50'50 MOUVMTS MAXI ADMIS
 POUR ATTEINDRE MAITRE
1020 REM HEURIST NAV (3)
1030 IF PXR<>PXM THEN NUME=ABS(PYR-PYM):
DENO=ABS(PXR-PXM):IN=NUME/DENO ELSE 1060
1040 IF IN<0.5 THEN VXR=SGN(PXM-PXR):VYR
=INT(3*RND-1)::GOTO 1070
1050 IF IN>2 THEN VXR=INT(3*RND-1)::VYR=
SGN(PYM-PYR):GOTO 1070
1060 VXR=SGN(PXM-PXR):VYR=SGN(PYM-PYR)
1070 TT=0'INITIALISATION DU RANG TT DES
TENTATIVES POUR TROUVER UNE POSITION LIB
RE APRES AVOIR HEURTE UN PILIER
1080 PXR=PPXR+VXR:PYR=PPYR+VYR
1090 IF PXR>32 THEN 1240 ELSE GOSUB 3000
'TEST HEURT PILIER
1100 IF OBS=0 THEN 1110 ELSE 1080'RECALC
UL POS
1110 IF RET=0 THEN 1120 ELSE 1080
1120 IF M<>MPI THEN 1240
1130 REM HEURIST DE DEGT (F):ESSAI AUTRE
 SENS D'EXPLORATION
1140 IF ESS=1 THEN 1180 ELSE PXR1=PXR:PY
R1=PYR:VXR1=VXR:VYR1=VYR'AFF ET MEMO 1ER
E POS
1150 SSS=SS:IF SSS=1 THEN SS=0 ELSE SS=1
'INVERSION SENS DE ROTATION
1160 TT=2:ESS=1:D=D1:GOSUB 3250
1170 GOTO 1080'RECALCUL POS
1180 ESS=0:MPI=0'REINIT ESS ET MPI
1190 REM HEUR DE DEGT (F):CHOIX MEILLEUR
E POSITION,PLUS COURTE DES 2 DISTANCES D
IST1 ET DIST2 AU MAITRE
1200 HOR1=ABS(PXM-PXR1):VERT1=ABS(PYM-PY
R1)::IF HOR1>VERT1 THEN DIST1=HOR1 ELSE
DIST1=VERT1
1210 HOR2=ABS(PXM-PXR):VERT2=ABS(PYM-PYR
):IF HOR2>VERT2 THEN DIST2=HOR2 ELSE DIS
T2=VERT2
1220 IF DIST2<=DIST1 THEN 1230 ELSE PXR=
PXR1:PYR=PYR1:VXR=VXR1:VYR=VYR1:GOTO 124
0'1ERE POS CHOISIE
1230 VVXR=VXR:VVYR=VYR'2EME POS CHOISIE
1240 PEN 3:LOCATE PPXR,PPYR:PRINT CHR$(3
3):PEN 1:LOCATE PXR,PYR:PRINT CHR$(33):S
OUND 1,30,5,7:PPXR=PXR:PPYR=PYR
1250 GOSUB 4000'TEST MAITRE REJOINT
1260 NEXT M
1270 SOUND 1,500,40,7:GOTO 4030'PARCOURS
 RATE
1280 END
2000 REM ZONE DES SOUS-PROGRAMMES
3000 REM SSPROG HEURT PILIER ET/OU RETOU
```

```
R EN ARRIERE
3010 OBS=0:RET=0
3020 REM TEST PILIER
3030 IF C(PXR,PYR)=1 THEN 3040 ELSE 3050
3040 OBS=1:GOTO 3080
3050 REM TEST RETOUR ARRIERE
3060 IF VXR<>-VVXR THEN RETURN ELSE IF V
YR<>-VVYR THEN RETURN
3070 IF TT>7 THEN RETURN ELSE RET=1
3080 TT=TT+1:MPI=M'MEMO NO BOUCLE
3090 REM ESSAI NOUVELLE DIRECTION
3100 REM HEURISTIQUE DE DEGAGEMENT (F):E
XPLORATION ORDONNEE DES  DIRECTIONS EN P
ARTANT DU PILIER HEURTE
3110 IF TT>1 THEN 3250'CAS 2EME PASSAGE
ET SUIVANTS
3120 IF VXR=-1 THEN 3150'CAS 1ER PASSAGE
,DETERMINATION DE LA DIRECTION D1 DU 1ER
 HEURT
3130 IF VXR=0 THEN 3180
3140 IF VXR =1 THEN 3200
3150 IF VYR=-1 THEN D1=5:GOTO 3230
3160 IF VYR=0 THEN D1=6:GOTO 3230
3170 IF VYR=1 THEN D1=7:GOTO 3230
3180 IF VYR=-1 THEN D1=4:GOTO 3230
3190 IF VYR=1 THEN D1=8:GOTO 3230
3200 IF VYR=-1 THEN D1=3:GOTO 3230
3210 IF VYR=0 THEN D1=2:GOTO 3230
3220 IF VYR=1 THEN D1=1
3230 D=D1'DIRECTION D DEBUT DE L'EXPLORA
TION
3240 SS=INT(2*RND)'TIRAGE SENS DE L'EXPL
ORATION
3250 REM EXPLORATION POSITION SUIVANTE
3260 IF SS=0 THEN D=D+1:IF D<9 THEN 3280
 ELSE D=D-8:GOTO 3280
3270 D=D-1:IF D>0 THEN 3280 ELSE D=D+8
3280 IF D=1 THEN 3290 ELSE IF D=2 THEN 3
300 ELSE IF D=3 THEN 3310 ELSE IF D=4 TH
EN 3320 ELSE IF D=5 THEN 3330 ELSE IF D=
6 THEN 3340 ELSE IF D=7 THEN 3350 ELSE 3
360
3290 VXR=1:VYR=1:GOTO 3370
3300 VXR=1:VYR=0:GOTO 3370
3310 VXR=1:VYR=-1:GOTO 3370
3320 VXR=0:VYR=-1:GOTO 3370
3330 VXR=-1:VYR=-1:GOTO 3370
3340 VXR=-1:VYR=0:GOTO 3370
3350 VXR=-1:VYR=1:GOTO 3370
3360 VXR=0:VYR=1
3370 RETURN
4000 REM SSPROG MAITRE REJOINT
4010 IF PXR<>PXM THEN RETURN ELSE IF PYR
<>PYM THEN RETURN
4020 FOR FOIS=1 TO 4:SOUND 1,27,5,7:FOR
```

```
AT=1 TO 100:NEXT:NEXT
4030 FOR AT=1 TO 800:NEXT:RG=RG+1:IF RG<
5 THEN PEN 3:LOCATE PPXR,PPYR:PRINT CHR$
(33):GOTO 180 ELSE 110
```

Commentaires techniques

Le flou est introduit dans les composantes de la vitesse du robot.

VI

CONCERTO ET VARIATIONS POUR DEUX ROBOTS

"Et que c'est long, le temps, et que c'est grand, le monde
Et que nous aurions pu ne pas nous rencontrer."
P. GÉRALDY.

Roméo va cesser de courir vers son maître. Il va maintenant, jusqu'en E20, s'élancer vers Juliette qui de son côté ira à sa rencontre avec le même empressement, c'est-à-dire, en langage artificiel, avec les mêmes heuristiques.

Ce scénario permettra aux humains, non seulement d'assister à quelques effusions d'Amour Artificiel (A.A.)[1], mais aussi d'étudier sérieusement un dispositif de commutation de la boucle des mouvements : il suffit ainsi d'une seule boucle pour les deux robots.

On trouvera par ailleurs en E17 une nouvelle heuristique originale de dégagement G et en E18 la première intervention de fonctions logiques pour représenter un univers plus riche.

1. Grâce à une onomatopée sophistiquée à la limite des possibilités du BASIC.

ÉTAPE E16

TOI ET MOI

Présentation

Le but principal de ce scénario est d'ordre technique : proposer un outil de programmation pour le mouvement alterné de deux robots utilisant les mêmes heuristiques de navigation et de dégagement, par exemple les heuristiques 1 et C.

Un tel dispositif, dont nous aurons besoin pour les quatre étapes suivantes, permet d'utiliser la même boucle pour les deux mobiles. Il comporte également un compteur de pas, affiché à l'écran, pour mesurer les performances, et le labyrinthe a maintenant 51 piliers.

Il est intéressant de noter l'impression de recul ressentie avec le retour de cette heuristique C après l'heuristique évoluée F de E13-E14. La sensation d'intelligence évoquée en E5, E11 puis E13 est donc relative ; on ressent le rapport des intelligences de deux programmes, plus que l'intelligence absolue d'un programme.

Telle heuristique brillamment ressentie donnera, quelque temps après, une piètre impression, confrontée à un récent perfectionnement vite entré dans les habitudes.

Programme

```
100 'ETAPE E16:DEUX ROBOTS SE CHERCHANT
 DANS LABYRINTHE,51 PILIERS,HEURISTIQUE
DE NAVIGATION (1),HEURISTIQUE DE DEGAGEM
ENT (C),5 PARCOURS SUR MEME TERRAIN,POSS
 CHANGMT TERRAIN
110 INK 0,16:INK 1,1:INK 2,6:INK 3,21:CL
S:CLEAR:RG=1
120 DIM C(32,25)'TABLEAU DES CASES DE L'
ECRAN SUSCEPTIBLES DE RECEVOIR UN PILIER
130 SYMBOL AFTER 32:SYMBOL 33,&0,&7C,&D6
,&6C,&38,&FE,&7C,&6C'DESSIN ROBOTS
140 'NOMBRES ALEATOIRES VARIES
150 RANDOMIZE TIME
160 CLS:H=INT(2*RND):IF H=1 THEN PXR1=IN
T(25*RND+6):PYR1=6 ELSE PXR1=6:PYR1=INT(
12*RND+6)'TIRAGE POSITION ROBOT NO 1
170 PXR2=36:PYR2=21'POSITION DEPART ROBO
T NO 2
```

```
180 PEN 2:LOCATE PXR1,PYR1:PRINT CHR$(33
):LOCATE PXR2,PYR2:PRINT CHR$(33)'AFF RO
BOTS AU DEPART
190 IF RG>1 THEN 270
200 FOR N=0 TO 50:I=INT((33-PXR1)*RND)+P
XR1-1:J=INT((21-PYR1)/(36-PXR1)*(I-PXR1)
)+INT((5*RND-2)*(I-PXR1)/8)+PYR1+1
210 IF J>PYR1-2 THEN 230
220 J=PYR1-1:GOTO 250
230 IF J<>PYR1 THEN IF J<22 THEN 250 ELS
E J=21:GOTO 250
240 IF I<>PXR1 THEN 250 ELSE J=PYR1-1
250 PEN 1:LOCATE I,J:PRINT"="'AFF PILIER

260 C(I,J)=1:NEXT'REPRESENTATION PILIER

270 PXA=PXR1:PYA=PYR1:PPXA=PXA:PPYA=PYA'
MISE EN BOUCLE DU ROBOT NO1 (ACTIF)
280 PXP=PXR2:PYP=PYR2'MISE EN BOUCLE DU
ROBOT NO2 (PASSIF)
290 SOUND 1,200,80,7:FOR AT=1 TO 2000:NE
XT'TOP DEPART
1000 REM BOUCLE DES MOUVEMENTS
1010 FOR M=1 TO 100
1020 TERR$=INKEY$:IF TERR$=" " THEN 110'
CHANGMT TERRAIN EVENTUEL
1030 PEN 1:LOCATE 31,2:PRINT M;" "'AFF C
OMPTEUR
1040 REM HEURIST NAV (1)
1050 IF PXA<>PXP THEN NUME=ABS(PYA-PYP):
DENO=ABS(PXA-PXP):IN=NUME/DENO ELSE 1080

1060 IF IN<0.5 THEN VXA=SGN(PXP-PXA):VYA
=0:GOTO 1090
1070 IF IN>2 THEN VXA=0:VYA=SGN(PYP-PYA)
:GOTO 1090
1080 VXA=SGN(PXP-PXA):VYA=SGN(PYP-PYA)
1090 JT=0'INITIALISATION DU RANG TT DES
TENTATIVES POUR TROUVER UNE POSITION LIB
RE APRES AVOIR HEURTE UN PILIER
1100 PXA=PPXA+VXA:PYA=PPYA+VYA
1110 IF PXA>31 THEN 1130 ELSE GOSUB 3000
1120 IF OBS=1 THEN 1100
1130 IF M/2=INT(M/2) THEN PER=30 ELSE PE
R=40
1140 PEN 3:LOCATE PPXA,PPYA:PRINT CHR$(3
3):PEN 2:LOCATE PXA,PYA:PRINT CHR$(33):S
OUND 1,PER,5,7:PPXA=PXA:PPYA=PYA
1150 GOSUB 4000'TEST RENCONTRE DES DEUX
ROBOTS
1160 REM INVERSION DES DEUX ROBOTS
1170 PXANOUV=PXP:PYANOUV=PYP:PXP=PXA:PYP
=PYA:PXA=PXANOUV:PYA=PYANOUV:PPXA=PXA:PP
YA=PYA
```

```
1180 FOR N=1 TO 80:NEXT:NEXT M
1190 SOUND 1,500,40,7:GOTO 4040'PARCOURS
 RATE
1200 END
2000 REM ZONE DES SOUS-PROGRAMMES
3000 REM SSPROG HEURT PILIER
3010 OBS=0:TT=TT+1
3020 IF C(PXA,PYA)<>1 THEN RETURN
3030 FOR AT=1 TO 150:NEXT:SOUND 1,24,5,7
::PEN 2:LOCATE PXA,PYA:PRINT"*":FOR AT=1
 TO 50:NEXT:PEN 1:LOCATE PXA,PYA:PRINT"=
"
3040 REM HEURIST DE DEGAGMT (C):TIRAGE A
U SORT 1ERE DIRECTION DE DEGAGEMENT,PUIS
,EN CAS D'INSUCCES,EXPLORATION ORDONNEE
ES AUTRES DIRECTIONS
3050 FOR AT=1 TO 80:NEXT:IF TT=1 THEN HA
S=INT(8*RND+1)::SS=INT(2*RND):GOTO 3090
3060 IF SS=0 THEN HAS=HAS+1::IF HAS<9 TH
EN 3090 ELSE HAS=HAS-8:GOTO 3090
3070 HAS=HAS-1:IF HAS>0 THEN 3090
3080 HAS=HAS+8
3090 IF HAS=1 THEN 3100 ELSE  IF HAS=2 T
HEN 3110 ELSE IF HAS=3 THEN 3120 ELSE IF
 HAS=4 THEN 3130 ELSE IF HAS=5 THEN 3140
 ELSE IF HAS=6 THEN 3150 ELSE IF HAS=7 T
HEN 3160 ELSE 3170
3100 VXA=1:VYA=1:GOTO 3180
3110 VXA=1:VYA=0:GOTO 3180
3120 VXA=1:VYA=-1:GOTO 3180
3130 VXA=0:VYA=-1:GOTO 3180
3140 VXA=-1:VYA=-1:GOTO 3180
3150 VXA=-1:VYA=0:GOTO 3180
3160 VXA=-1:VYA=1:GOTO 3180
3170 VXA=0:VYA=1
3180 OBS=1
3200 RETURN
4000 REM SSPROG RENCONTRE DES DEUX ROBOT
S
4010 IF PXA<>PXP THEN RETURN ELSE IF PYA
<>PYP THEN RETURN
4020 PEN 1:LOCATE PXA,PYA:PRINT CHR$(33)
4030 FOR FOIS=1 TO 4:SOUND 1,27,5,7:FOR
AT=1 TO 100:NEXT:NEXT
4040 FOR AT=1 TO 800:NEXT:RG=RG+1:IF RG<
6 THEN PEN 3:LOCATE PXA,PYA:PRINT CHR$(3
3):GOTO 180 ELSE 110
```

Commentaires techniques

- Le principe consiste à rendre alternativement actif puis passif chacun

des deux protagonistes. Chacun exécute un pas en prenant l'autre comme cible, puis sert lui-même de cible à son partenaire.

- C'est le robot n° 1 (Juliette) situé à gauche qui fait le premier pas, conformément aux usages des pays situés à la pointe de la robotique. Juliette est donc "mise en boucle" en tant que robot actif A. Roméo est de même mis en boucle en tant que robot passif P.
- On ne trouvera ensuite dans la boucle des mouvements que ces deux "robots de paille" qui changeront d'identité à chaque nouvelle boucle.
- Cette boucle est exactement rédigée comme d'habitude, en remplaçant le maître M par le robot passif P et le robot R par le robot actif A. Seule la tonalité du bip ponctuant le pas de chaque robot est différenciée pour faciliter le suivi. On utilise à cette fin la parité ou l'imparité du rang de la boucle (impair pour Juliette, pair pour Roméo).
- En fin de boucle on opère la commutation, c'est-à-dire l'inversion des deux robots, A prenant la place de P et réciproquement. On notera l'utilisation des variables accessoires PXANOUV et PYANOUV pour éviter un cercle vicieux dans les échanges d'identité. Le lecteur perplexe pourra essayer de s'affranchir de cette complication apparemment gratuite pour se convaincre de son utilité.
- Les sous-programmes sont également identiques à ceux des exemples précédents.

ÉTAPE E17

UNE HEURISTIQUE PIPÉE POUR UN PAS DE DEUX

Présentation

Le labyrinthe comporte donc 51 piliers et non plus les 31 habituels. Leur densité complique ainsi la tâche des robots et les amène à utiliser les grands moyens : *tricher*.

C'est ainsi qu'avec une heuristique de dégagement relativement médiocre, l'heuristique C de E16, ils vont tenter de se retrouver plus rapidement, en effectuant un pas de deux à pirouettes, plus précisément en effectuant deux pas dans la direction de dégagement au lieu d'un.

Cette heuristique malhonnête prend l'appellation G.

Les parcours successifs obtenus montrent l'efficacité du procédé : lorsqu'il s'agit de s'écarter d'un danger, deux fois valent mieux qu'une...

Programme

```
100 'ETAPE E17:DEUX ROBOTS SE CHERCHANT
 DANS LABYRINTHE,51 PILIERS,HEURISTIQUE
DE NAVIGATION (1),HEURISTIQUE DE DEGAGEM
ENT (G),5 PARCOURS SUR MEME TERRAIN,POSS
 CHANGMT TERRAIN
110 INK 0,16:INK 1,1:INK 2,6:INK 3,21:CL
S:CLEAR:RG=1
120 DIM C(32,25)'TABLEAU DES CASES DE L'
ECRAN SUSCEPTIBLES DE RECEVOIR UN PILIER
130 SYMBOL AFTER 32:SYMBOL 33,&0,&7C,&D6
,&6C,&38,&FE,&7C,&6C'DESSIN ROBOTS
140 'NOMBRES ALEATOIRES VARIES
150 RANDOMIZE TIME
160 CLS:H=INT(2*RND):IF H=1 THEN PXR1=IN
T(25*RND+6):PYR1=6 ELSE PXR1=6:PYR1=INT(
12*RND+6)'TIRAGE POSITION ROBOT NO 1
170 PXR2=36:PYR2=21'POSITION DEPART ROBO
T NO 2
180 PEN 2:LOCATE PXR1,PYR1:PRINT CHR$(33
):LOCATE PXR2,PYR2:PRINT CHR$(33)'AFF RO
BOTS AU DEPART
190 IF RG>1 THEN 270
200 FOR N=0 TO 50:I=INT((33-PXR1)*RND)+P
XR1-1:J=INT((21-PYR1)/(36-PXR1)*(I-PXR1)
)+INT((5*RND-2)*(I-PXR1)/8)+PYR1+1
210 IF J>PYR1-2 THEN 230
```

```
220 J=PYR1-1:GOTO 250
230 IF J<>PYR1 THEN IF J<22 THEN 250 ELS
E J=21:GOTO 250
240 IF I<>PXR1 THEN 250 ELSE J=PYR1-1
250 PEN 1:LOCATE I,J:PRINT"="'AFF PILIER

260 C(I,J)=1:NEXT'REPRESENTATION PILIER

270 PXA=PXR1:PYA=PYR1:PPXA=PXA:PPYA=PYA'
MISE EN BOUCLE DU ROBOT NO1 (ACTIF)
280 PXP=PXR2:PYP=PYR2'MISE EN BOUCLE DU
ROBOT NO2 (PASSIF)
290 SOUND 1,200,80,7:FOR AT=1 TO 2000:NE
XT'TOP DEPART
295 MOB=-2'INITIALISATION PARAMETRE MOB
POUR EVITER DEVOIEMENTS INTEMPESTIFS AUX
 BOUCLES NOS 1 ET 2
1000 REM BOUCLE DES MOUVEMENTS
1010 FOR M=1 TO 100
1020 TERR$=INKEY$:IF TERR$=" " THEN 110'
CHANGMT TERRAIN EVENTUEL
1030 PEN 1:LOCATE 31,2:PRINT M;" "'AFF C
OMPTEUR
1040 REM HEURIST NAV (1)
1050 REM HEURIST DE DEGMT (G):SI HEURT A
 BOUCLE M=MOB,PAS DE MOUVMT DE L'AUTRE R
OBOT A BOUCLE MOB+1,ET REPETITIONDU MOUV
MT DU ROBOT HEURTEUR A BOUCLE MOB+2
1060 IF M=MOB+1 THEN 1190
1070 IF M=MOB+2 THEN 1130
1080 IF PXA<>PXP THEN NUME=ABS(PYA-PYP):
DENO=ABS(PXA-PXP):IN=NUME/DENO ELSE 1110

1090 IF IN<0.5 THEN VXA=SGN(PXP-PXA):VYA
=0:GOTO 1120
1100 IF IN>2 THEN VXA=0:VYA=SGN(PYP-PYA)
:GOTO 1120
1110 VXA=SGN(PXP-PXA):VYA=SGN(PYP-PYA)
1120 TT=0'INITIALISATION DU RANG TT DES
TENTATIVES POUR TROUVER UNE POSITION LIB
RE APRES AVOIR HEURTE UN PILIER
1130 PXA=PPXA+VXA:PYA=PPYA+VYA
1140 IF PXA>31 THEN 1160 ELSE GOSUB 3000
1150 IF OBS=1 THEN 1130
1160 IF M/2=INT(M/2) THEN PER=30 ELSE PE
R=40
1170 PEN 3:LOCATE PPXA,PPYA:PRINT CHR$(3
3):PEN 2:LOCATE PXA,PYA:PRINT CHR$(33):S
OUND 1,PER,5,7:PPXA=PXA:PPYA=PYA
1180 GOSUB 4000'TEST RENCONTRE DES DEUX
ROBOTS
1190 REM INVERSION DES DEUX ROBOTS
1200 PXANOUV=PXP:PYANOUV=PYP:PXP=PXA:PYP
=PYA:PXA=PXANOUV:PYA=PYANOUV:PPXA=PXA:PP
```

```
YA=PYA
1210 FOR N=1 TO 80:NEXT:NEXT M
1220 SOUND 1,500,40,7:GOTO 4040'PARCOURS
 RATE
1230 END
2000 REM ZONE DES SOUS-PROGRAMMES
3000 REM SSPROG HEURT PILIER
3010 OBS=0:TT=TT+1
3020 IF C(PXA,PYA)<>1 THEN RETURN
3030 FOR AT=1 TO 150:NEXT:SOUND 1,24,5,7
::PEN 2:LOCATE PXA,PYA:PRINT"*":FOR AT=1
 TO 50:NEXT:PEN 1:LOCATE PXA,PYA:PRINT"=
"
3040 REM HEURIST DE DEGAGMT (G):TIRAGE A
U SORT 1ERE DIRECTION DE DEGAGEMENT,PUIS
,EN CAS D'INSUCCES,EXPLORATION ORDONNEE
ES AUTRES DIRECTIONS
3050 FOR AT=1 TO 80:NEXT:IF TT=1 THEN HA
S=INT(8*RND+1)::SS=INT(2*RND):GOTO 3090
3060 IF SS=0 THEN HAS=HAS+1::IF HAS<9 TH
EN 3090 ELSE HAS=HAS-8:GOTO 3090
3070 HAS=HAS-1:IF HAS>0 THEN 3090
3080 HAS=HAS+8
3090 IF HAS=1 THEN 3100 ELSE  IF HAS=2 T
HEN 3110 ELSE IF HAS=3 THEN 3120 ELSE IF
 HAS=4 THEN 3130 ELSE IF HAS=5 THEN 3140
 ELSE IF HAS=6 THEN 3150 ELSE IF HAS=7 T
HEN 3160 ELSE 3170
3100 VXA=1:VYA=1:GOTO 3180
3110 VXA=1:VYA=0:GOTO 3180
3120 VXA=1:VYA=-1:GOTO 3180
3130 VXA=0:VYA=-1:GOTO 3180
3140 VXA=-1:VYA=-1:GOTO 3180
3150 VXA=-1:VYA=0:GOTO 3180
3160 VXA=-1:VYA=1:GOTO 3180
3170 VXA=0:VYA=1
3180 OBS=1:MOB=M'MEMO NO BOUCLE
3190 RETURN
4000 REM SSPROG RENCONTRE DES DEUX ROBOT
S
4010 IF PXA<>PXP THEN RETURN ELSE IF PYA
<>PYP THEN RETURN
4020 PEN 1:LOCATE PXA,PYA:PRINT CHR$(33)
4030 FOR FOIS=1 TO 4:SOUND 1,27,5,7:FOR
AT=1 TO 100:NEXT:NEXT
4040 FOR AT=1 TO 800:NEXT:RG=RG+1:IF RG<
6 THEN PEN 3:LOCATE PXA,PYA:PRINT CHR$(3
3):GOTO 180 ELSE 110
```

Commentaires techniques

- Le double dégagement est obtenu en retouchant légèrement la boucle des mouvements.

- Si un robot rencontre un pilier lors de la boucle de rang M (rang que l'on mémorise dans le sous-programme heurt pilier sous l'identité MOB = M), l'autre robot ne fait aucun mouvement lors de la boucle suivante MOB + 1, et le premier cité répète son mouvement de dégagement lors de la boucle MOB + 2.
- Noter qu'avant le départ de chaque parcours, il faut initialiser ou réinitialiser le paramètre MOB à la valeur bizarre de − 2 pour éviter des anomalies aux boucles n os 1 et 2 du premier parcours, puis aux boucles MOB + 1 et MOB + 2 correspondant au dernier pilier rencontré lors du parcours précédent.

ÉTAPE E18

DIVERSITÉ QUALITATIVE — LES FONCTIONS LOGIQUES

Présentation

Jusqu'à présent, le monde des robots se réduisait à des piliers. Cet univers ultra-simplifié était facilement représentable dans l'ordinateur, soit par les coordonnées de ces piliers, soit, à partir de E9, par la vérité ou la fausseté d'un prédicat affirmant : ''Dans telle case de l'écran, il y a un pilier'', et prenant conventionnellement la valeur 1 dans le premier cas et 0 dans le second. Nous avons pu constater la rapidité d'exécution que procure cet outil à deux positions.

Cette fois, le monde se complique : l'architecte Dédale a placé dans son labyrinthe, non pas un Minotaure, mais de visqueuses créatures, les Gluons, que Roméo et Juliette doivent dévorer dès qu'ils en touchent une, avant de poursuivre leur chemin.

Pour représenter cet univers un peu plus complexe, nous allons utiliser une entité C(I, J) susceptible de prendre, non plus seulement les deux valeurs 0 ou 1 (vrai ou faux) d'un prédicat, mais une infinité de valeurs possibles, suivant ce qu'elle doit désigner.

Ces valeurs peuvent être numériques (0, 1, 2, 3...), mais elles peuvent tout aussi bien être des chaînes de caractères (des mots) et c'est cette dernière forme que nous allons tester dans le présent exemple.

Cette entité, de portée plus générale que le prédicat, prend le nom de *fonction logique*. Ici, nous l'appellerons C$(I, J) pour marquer sa nature non numérique. Elle prend pour chaque case d'écran de coordonnées I, J une valeur (une appellation qualitative) qui varie suivant la nature de l'élément qui se touve dans la case.

Ainsi, l'on décide que :

- si la case est vide, C$(I, J) = '' '' (chaîne de caractères ''vide'') ;
- si la case contient un pilier, C$(I, J) = ''PIL'' ;
- si la case contient un Gluon (obstacle éliminable), C$(I, J) = ''OE'' ;
- si la case contient le robot passif, C$(I, J) = ''RO''.

Rappelons, suivant la terminologie adoptée en E16, que le robot actif est celui qui fait le mouvement élémentaire tandis que l'autre robot, dit passif, reste immobile, et qu'à la boucle suivante les rôles sont inversés.

La fonction logique C$(I, J) représente donc tous les éléments de l'univers du robot actif, éléments vis-à-vis desquels il a des comportements spécifiques différents : il *passe* dans les cases vides, il *contourne* les piliers, il *dévore* les Gluons et il *termine* son parcours s'il atteint le robot passif.

À l'exception du premier qui n'en est pas un, chacun de ces trois comportements est réglé par un sous-programme spécialisé, ainsi que nous l'avons déjà constaté dans des cas plus simples.

Pour savoir s'il existe ou non "quelque chose" dans la case de coordonnées PXA, PYA où se trouve le robot actif, il suffit de voir quelle valeur (appellation) prend la fonction C$(PXA, PYA).

C'est simple, général et puissant.

L'heuristique de dégagement C adoptée dans le présent exemple est assez simple, pour ne pas surcharger le programme démonstratif.

L'on voit Roméo et Juliette cheminer et dévorer les Gluons par des itinéraires divers. Lors de chaque parcours, le terrain est progressivement nettoyé des Gluons qui l'encombrent et nous verrons dans les commentaires techniques comment leur disparition se traduit dans le programme.

Le nombre de piliers a été multiplié pour accentuer la variété des itinéraires ; chaque terrain nouveau est parcouru cinq fois.

Programme

```
100 'ETAPE E18:2 ROB DS LAB,PIL ET OBST
ELIMINABLES,HEUR NAV (1),HEUR DEG (C),UT
ILISATION FONCTION LOGIQUE C$(I,J) POUR
REPR LES ELEM DE L'UNIVERS DU ROB ACTIF,
5 PARC SR MM TERR,POSS CHGMT TERR
110 INK 0,16:INK 1,1:INK 2,6:INK 3,21:CL
S:CLEAR:RG=1
120 REM FONCTION LOGIQUE C$(I,J) DONNANT
 L'OCCUPATION DES DIVERSES CASES DE L'EC
RAN
130 REM VALEURS DE C$ POUR LES DIVERS EL
EMENTS ELEMENTS DE L'UNIVERS DU ROBOT AC
TIF;CASE VIDE:C$=" ";PILIER:C$="PI";OBST
 ELIMINABLE:C$="OE";ROBOT PASSIF:C$="RO"
140 DIM C$(35,25)'TABLEAU DES CASES DE L
'ECRAN SUSCEPTIBLES DE RECEVOIR UN ELEME
NT
150 SYMBOL AFTER 32:SYMBOL 33,&0,&7C,&D6
,&6C,&38,&FE,&7C,&6C'DESSIN ROBOTS
160 'NOMBRES ALEATOIRES VARIES
170 RANDOMIZE TIME
180 CLS:PXR1=5:PYR1=INT(15*RND+6)'TIRAGE
 POS ROBOT NO 1
```

```
190 PXR2=35:PYR2=INT(15*RND+6)'TIRAGE PO
S ROBOT NO 2
200 PEN 1:LOCATE PXR1,PYR1:PRINT CHR$(33
):LOCATE PXR2,PYR2:PRINT CHR$(33)'AFF RO
BOTS AU DEPART
210 IF RG>1 THEN 280
220 FOR N=1 TO 100:I=2*INT(12*RND)+6:J=I
NT(17*RND+5)'TIRAGE POS 100 PILIERS SUR
COLONNES PAIRES
230 PEN 1:LOCATE I,J:PRINT"="'AFF PILIER
240 C$(I,J)="PI":NEXT 'REPRESENTATION PI
LIER
250 FOR N=1 TO 100:I=2*INT(12*RND)+7:J=I
NT(19*RND+4)'TIRAGE POS 100 OBSTACLES EL
IMINABLES SUR COLONNES IMPAIRES
260 PEN 2:LOCATE I,J:PRINT CHR$(189)'AFF
 OBST ELIMà
270 C$(I,J)="OE"::NEXT'REPRES OBST ELIM
280 PXA=PXR1:PYA=PYR1:PPXA=PXA:PPYA=PYA'
MISE EN BOUCLE DU ROBOT NO1 (ACTIF)
290 PXP=PXR2:PYP=PYR2'MISE EN BOUCLE DU
ROBOT NO2 (PASSIF)
300 C$(PXP,PYP)="RO"'REPRESENTATION ROBO
T PASSIF
310 SOUND 1,200,80,7:FOR AT=1 TO 2000:NE
XT'TOP DEPART
1000 REM BOUCLE DES MOUVEMENTS
1010 FOR M=1 TO 200
1020 TERR$=INKEY$:IF TERR$=" " THEN 110'
CHANGMT TERRAIN EVENTUEL
1030 PEN 1:LOCATE 31,2:PRINT M;" "'AFF C
OMPTEUR
1040 REM HEURIST NAV (1)
1050 IF PXA<>PXP THEN NUME=ABS(PYA-PYP):
DENO=ABS(PXA-PXP):IN=NUME/DENO ELSE 1080

1060 IF IN<0.5 THEN VXA=SGN(PXP-PXA):VYA
=0:GOTO 1090
1070 IF IN>2 THEN VXA=0:VYA=SGN(PYP-PYA)
:GOTO 1090
1080 VXA=SGN(PXP-PXA):VYA=SGN(PYP-PYA)
1090 TT=0'INITIALISATION DU RANG TT DES
TENTATIVES POUR TROUVER UNE POSITION LIB
RE APRES AVOIR HEURTE UN PILIER
1100 PXA=PPXA+VXA:PYA=PPYA+VYA
1110 IF PXA>29 THEN 1130 ELSE GOSUB 3000
'TEST HEURT PILIER
1120 IF OBS=1 THEN 1100
1130 IF PXA>29 THEN 1140 ELSE GOSUB 4000
'TEST RENCONTRE OBST ELIM
1140 IF M/2=INT(M/2) THEN PER=30 ELSE PE
R=40
1150 PEN 3:LOCATE PPXA,PPYA:PRINT CHR$(3
3):PEN 1:LOCATE PXA,PYA:PRINT CHR$(33):S
```

```
OUND 1,PER,5,7:PPXA=PXA:PPYA=PYA
1160 GOSUB 5000'TEST RENCONTRE DES DEUX
ROBOTS
1170 REM INVERSION DES DEUX ROBOTS
1180 PXANOUV=PXP:PYANOUV=PYP:PXP=PXA:PYP
=PYA:PXA=PXANOUV:PYA=PYANOUV:PPXA=PXA:PP
YA=PYA
1190 REM INVERSION REPRESENTATION DU ROB
OT PASSIF
1200 C$(PXA,PYA)=" ":C$(PXP,PYP)="RO"
1210 FOR N=1 TO 300:NEXT:NEXT M
1220 SOUND 1,500,40,7:GOTO 5040'PARCOURS
 RATE
1230 END
2000 REM ZONE DES SOUS-PROGRAMMES
3000 REM SSPROG HEURT PILIER
3010 OBS=0:TT=TT+1
3020 IF C$(PXA,PYA)<>"PI" THEN RETURN
3030 FOR AT=1 TO 150:NEXT:SOUND 1,24,5,7
::PEN 2:LOCATE PXA,PYA:PRINT"*":FOR AT=1
 TO 50:NEXT:PEN 1:LOCATE PXA,PYA:PRINT"=
"
3040 REM HEURIST DE DEGAGMT (C):TIRAGE A
U SORT 1ERE DIRECTION DE DEGAGEMENT,PUIS
,EN CAS D'INSUCCES,EXPLORATION ORDONNEE
ES AUTRES DIRECTIONS
3050 FOR AT=1 TO 80:NEXT:IF TT=1 THEN HA
S=INT(8*RND+1)::SS=INT(2*RND):GOTO 3090
3060 IF SS=0 THEN HAS=HAS+1::IF HAS<9 TH
EN 3090 ELSE HAS=HAS-8:GOTO 3090
3070 HAS=HAS-1:IF HAS>0 THEN 3090
3080 HAS=HAS+8
3090 IF HAS=1 THEN 3100 ELSE  IF HAS=2 T
HEN 3110 ELSE IF HAS=3 THEN 3120 ELSE IF
 HAS=4 THEN 3130 ELSE IF HAS=5 THEN 3140
 ELSE IF HAS=6 THEN 3150 ELSE IF HAS=7 T
HEN 3160 ELSE 3170
3100 VXA=1:VYA=1:GOTO 3180
3110 VXA=1:VYA=0:GOTO 3180
3120 VXA=1:VYA=-1:GOTO 3180
3130 VXA=0:VYA=-1:GOTO 3180
3140 VXA=-1:VYA=-1:GOTO 3180
3150 VXA=-1:VYA=0:GOTO 3180
3160 VXA=-1:VYA=1:GOTO 3180
3170 VXA=0:VYA=1
3180 OBS=1
3190 RETURN
4000 REM SSPROG RENCONTRE OBST ELIMINABL
E
4010 IF C$(PXA,PYA)<>"OE" THEN RETURN
4020 FOR AT=1 TO 150:NEXT:SOUND 1,24,5,7
:PEN 1:LOCATE PXA,PYA:PRINT"*":FOR AT=1
TO 50:NEXT:PEN 2:LOCATE PXA,PYA:PRINT CH
R$(189)'VISUALISATION RENCONTRE
```

```
4030 REM ELIMINATION OBSTACLE
4040 :C$(PXA,PYA)=" ":SOUND 1,2000,30,7:
SOUND 1,700,20,7:FOR AT=1 TO 300:NEXT:RE
TURN
5000 REM SSPROG RENCONTRE DES DEUX ROBOT
S
5010 IF PXA<>PXP THEN RETURN ELSE IF PYA
<>PYP THEN RETURN
5020 PEN 1:LOCATE PXA,PYA:PRINT CHR$(33)
5030 FOR FOIS=1 TO 4:SOUND 1,27,5,7:FOR
AT=1 TO 100:NEXT:NEXT
5040 FOR AT=1 TO 800:NEXT:RG=RG+1:IF RG<
6 THEN PEN 3:LOCATE PXA,PYA:PRINT CHR$(3
3):GOTO 200 ELSE 110
```

Commentaires techniques

- Les divers éléments sont tirés au sort, affichés à l'écran, puis représentés par la fonction C$(I, J), en amont de la boucle des mouvements.
- On voit, dans les trois sous-programmes spécialisés, l'intervention de la fonction logique C$ pour tester la présence éventuelle d'un pilier, d'un Gluon ou du robot passif dans la case où se trouve le robot actif.
- Noter que les Gluons, après avoir été dévorés, doivent bien entendu disparaître de la scène, visuellement et sémantiquement, comme nous l'avions déjà vu avec les prédicats. Cette fois, la disparition sémantique se traduit par la remise à vide de la fonction C$(PXA, PYA).

ÉTAPE E19

DES ROBOTS QUI BROIENT DU NOIR — LE TEST DE TURING

Présentation

En regardant Roméo et Juliette parcourir plus ou moins maladroitement le labyrinthe, on est tenté de penser qu'ils ne sont pas très futés et que l'on ferait certainement mieux.

Rappelons à ce propos que l'Anglais Alan Turing a inventé un test célèbre consistant, pour un observateur humain, à comparer dans des conditions parfaites d'"anonymat" les performances d'un programme d'I.A. et celles d'un homme.

Si l'observateur n'arrive pas à faire la différence, on en déduit que le programme est intelligent. On se garde bien de déduire que l'homme est idiot, car une telle hypothèse n'a jamais effleuré personne.

Si l'observateur était un programme d'I.A., il déduirait probablement que l'homme est intelligent pour les mêmes raisons d'autoréférence que ci-dessus.

Redevenons sérieux [2] et revenons à nos deux robots mal dégrossis.

Il est difficile de se rendre compte que cette fâcheuse impression provient en grande partie du fait que le spectateur humain *voit* les positions d'ensemble et de détail des piliers et des Gluons, alors que les robots, comme nous l'avons précisé, ne disposent que de leur sens du toucher, leur vue leur permettant seulement de situer la direction de leur coreligionnaire.

Un sens supplémentaire constitue une importante source de valeur ajoutée pour l'intelligence. Il est facile de s'en rendre compte en privant le spectateur de cet atout : tel est l'objet du présent exemple.

Après la mise en place habituelle des piliers et des Gluons de E18, l'écran s'obscurcit et l'on ne voit plus, dans le noir absolu, que Roméo et Juliette. Ils n'en sont nullement gênés et entament leur périple en opérant comme d'habitude, c'est-à-dire en tâtonnant, ce qui est aussi efficace dans le noir qu'à la lumière du jour. Il n'en est pas de même pour le spectateur, alors bien obligé de se mettre à leur place, et qui devient ainsi un peu plus indulgent, sinon plus admiratif.

---

2. Le test de Turing est brillamment présenté, avec toutes ses subtilités, dans le livre de Douglas Hofstadter cité au Chapitre III.

Programme

```
100 'ETAPE E19:COMME E18 AVEC ECRAN OBSC
UR
110 INK 0,16:INK 1,1:INK 2,6:INK 3,21:PA
PER 0:CLS:CLEAR:RG=1
120 REM FONCTION LOGIQUE C$(I,J) DONNANT
 L'OCCUPATION DES DIVERSES CASES DE L'EC
RAN
130 REM VALEURS DE C$ POUR LES DIVERS EL
EMENTS ELEMENTS DE L'UNIVERS DU ROBOT AC
TIF;CASE VIDE:C$=" ";PILIER:C$="PI";OBST
 ELIMINABLE:C$="OE";ROBOT PASSIF:C$="RO"
140 DIM C$(35,25)'TABLEAU DES CASES DE L
'ECRAN SUSCEPTIBLES DE RECEVOIR UN ELEME
NT
150 SYMBOL AFTER 32:SYMBOL 33,&0,&7C,&D6
,&6C,&38,&FE,&7C,&6C'DESSIN ROBOTS
160 'NOMBRES ALEATOIRES VARIES
170 RANDOMIZE TIME
180 CLS:PXR1=5:PYR1=INT(15*RND+6)'TIRAGE
 POS ROBOT NO 1
190 PXR2=35:PYR2=INT(15*RND+6)'TIRAGE PO
S ROBOT NO 2
200 PEN 2:LOCATE PXR1,PYR1:PRINT CHR$(33
):LOCATE PXR2,PYR2:PRINT CHR$(33)'AFF RO
BOTS AU DEPART
210 IF RG>1 THEN 280
220 FOR N=1 TO 100:I=2*INT(12*RND)+6:J=I
NT(17*RND+5)'TIRAGE POS 100 PILIERS SUR
COLONNES PAIRES
230 PEN 1:LOCATE I,J:PRINT"="'AFF PILIER
240 C$(I,J)="PI":NEXT 'REPRESENTATION PI
LIER
250 FOR N=1 TO 100:I=2*INT(12*RND)+7:J=I
NT(19*RND+4)'TIRAGE POS 100 OBSTACLES EL
IMINABLES SUR COLONNES IMPAIRES
260 PEN 2:LOCATE I,J:PRINT CHR$(189)'AFF
 OBST ELIMà
270 C$(I,J)="OE"::NEXT'REPRES OBST ELIM
280 PXA=PXR1:PYA=PYR1:PPXA=PXA:PPYA=PYA'
MISE EN BOUCLE DU ROBOT NO1 (ACTIF)
290 PXP=PXR2:PYP=PYR2'MISE EN BOUCLE DU
ROBOT NO2 (PASSIF)
300 C$(PXP,PYP)="RO"'REPRESENTATION ROBO
T PASSIF
310 IF RG<>1 THEN 330 ELSE FOR AT=1 TO 5
00:NEXT:SOUND 1,1000,60,7:FOR AT=1 TO 80
0:NEXT:PAPER 1:CLS'OBSCURCISSEMENT DE L'
ECRAN
320 FOR AT=1 TO 400:NEXT:PEN 2:LOCATE PX
A,PYA:PRINT CHR$(33):LOCATE PXP,PYP:PRIN
T CHR$(33)'REAFF ROBOTS AU DEPART
330 SOUND 1,200,80,7:FOR AT=1 TO 2000:NE
```

```
XT'TOP DEPART
1000 REM BOUCLE DES MOUVEMENTS
1010 FOR M=1 TO 200
1020 TERR$=INKEY$:IF TERR$=" " THEN 110'
CHANGMT TERRAIN EVENTUEL
1030 PEN 1:LOCATE 31,2:PRINT M;" "'AFF C
OMPTEUR
1040 REM HEURIST NAV (1)
1050 IF PXA<>PXP THEN NUME=ABS(PYA-PYP):
DENO=ABS(PXA-PXP):IN=NUME/DENO ELSE 1080
1060 IF IN<0.5 THEN VXA=SGN(PXP-PXA):VYA
=0:GOTO 1090
1070 IF IN>2 THEN VXA=0:VYA=SGN(PYP-PYA)
:GOTO 1090
1080 VXA=SGN(PXP-PXA):VYA=SGN(PYP-PYA)
1090 TT=0'INITIALISATION DU RANG TT DES
TENTATIVES POUR TROUVER UNE POSITION LIB
RE APRES AVOIR HEURTE UN PILIER
1100 PXA=PPXA+VXA:PYA=PPYA+VYA
1110 IF PXA>29 THEN 1130 ELSE GOSUB 3000
'TEST HEURT PILIER
1120 IF OBS=1 THEN 1100
1130 IF PXA>29 THEN 1140 ELSE GOSUB 4000
'TEST RENCONTRE OBST ELIM
1140 IF M/2=INT(M/2) THEN PER=30 ELSE PE
R=40
1150 PEN 3:LOCATE PPXA,PPYA:PRINT CHR$(3
3):PEN 2:LOCATE PXA,PYA:PRINT CHR$(33):S
OUND 1,PER,5,7:PPXA=PXA:PPYA=PYA
1160 GOSUB 5000'TEST RENCONTRE DES DEUX
ROBOTS
1170 REM INVERSION DES DEUX ROBOTS
1180 PXANOUV=PXP:PYANOUV=PYP:PXP=PXA:PYP
=PYA:PXA=PXANOUV:PYA=PYANOUV:PPXA=PXA:PP
YA=PYA
1190 REM INVERSION REPRESENTATION DU ROB
OT PASSIF
1200 C$(PXA,PYA)=" ":C$(PXP,PYP)="RO"
1210 FOR N=1 TO 300:NEXT:NEXT M
1220 SOUND 1,500,40,7:GOTO 5040'PARCOURS
 RATE
1230 END
2000 REM ZONE DES SOUS-PROGRAMMES
3000 REM SSPROG HEURT PILIER
3010 OBS=0:TT=TT+1
3020 IF C$(PXA,PYA)<>"PI" THEN RETURN
3030 FOR AT=1 TO 150:NEXT:SOUND 1,24,5,7
::PEN 2:LOCATE PXA,PYA:PRINT"=":FOR AT=1
 TO 50:NEXT:PEN 1:LOCATE PXA,PYA:PRINT"=
"
3040 REM HEURIST DE DEGAGMT (C):TIRAGE A
U SORT 1ERE DIRECTION DE DEGAGEMENT,PUIS
,EN CAS D'INSUCCES,EXPLORATION ORDONNEE
ES AUTRES DIRECTIONS
```

```
3050 FOR AT=1 TO 80:NEXT:IF TT=1 THEN HA
S=INT(8*RND+1)::SS=INT(2*RND):GOTO 3090
3060 IF SS=0 THEN HAS=HAS+1::IF HAS<9 TH
EN 3090 ELSE HAS=HAS-8:GOTO 3090
3070 HAS=HAS-1:IF HAS>0 THEN 3090
3080 HAS=HAS+8
3090 IF HAS=1 THEN 3100 ELSE  IF HAS=2 T
HEN 3110 ELSE IF HAS=3 THEN 3120 ELSE IF
 HAS=4 THEN 3130 ELSE IF HAS=5 THEN 3140
 ELSE IF HAS=6 THEN 3150 ELSE IF HAS=7 T
HEN 3160 ELSE 3170
3100 VXA=1:VYA=1:GOTO 3180
3110 VXA=1:VYA=0:GOTO 3180
3120 VXA=1:VYA=-1:GOTO 3180
3130 VXA=0:VYA=-1:GOTO 3180
3140 VXA=-1:VYA=-1:GOTO 3180
3150 VXA=-1:VYA=0:GOTO 3180
3160 VXA=-1:VYA=1:GOTO 3180
3170 VXA=0:VYA=1
3180 OBS=1
3190 RETURN
4000 REM SSPROG RENCONTRE OBST ELIMINABL
E
4010 IF C$(PXA,PYA)<>"OE" THEN RETURN
4020 FOR AT=1 TO 150:NEXT:SOUND 1,24,5,7
:PEN 2:LOCATE PXA,PYA:PRINT CHR$(189)'VI
SUALISATION RENCONTRE
4030 REM ELIMINATION OBSTACLE
4040 :C$(PXA,PYA)=" ":FOR AT=1 TO 800:NE
XT:SOUND 1,2000,30,7:SOUND 1,200,5,7:FOR
 AT=1 TO 500:NEXT:RETURN
5000 REM SSPROG RENCONTRE DES DEUX ROBOT
S
5010 IF PXA<>PXP THEN RETURN ELSE IF PYA
<>PYP THEN RETURN
5020 PEN 0:LOCATE PXA,PYA:PRINT CHR$(33)
5030 FOR FOIS=1 TO 4:SOUND 1,27,5,7:FOR
AT=1 TO 100:NEXT:NEXT
5040 FOR AT=1 TO 1500:NEXT:RG=RG+1:IF RG
<6 THEN PEN 3:LOCATE PXA,PYA:PRINT CHR$(
33):GOTO 200 ELSE 110
```

Commentaires techniques

- L'écran ne s'obscurcit qu'après l'affichage de tous les éléments, pour mettre le spectateur en condition.
- L'accompagnement audiovisuel des tâtonnements des robots en mouvement est conservé, et prend ici toute sa valeur.

ÉTAPE E20

LE FIL D'ARIANE

Présentation

Même scénario qu'en E16, Roméo et Juliette se cherchent dans le labyrinthe désert.

Mais, fatigués de parcourir plusieurs fois le même terrain et de toujours procéder à de fastidieux tâtonnements, ils décident de se concentrer, de mémoriser avec précision leur premier parcours afin de le refaire ensuite sans hésiter et sans toucher un seul pilier.

La mémorisation des positions successives est réalisée dans un tableau indicé que le BASIC permet de construire, puis de manipuler aisément. Le programme ci-après permet d'apprécier l'effet assez spectaculaire de cette restitution mentale.

Programme

```
100 'ETAPE E20:DEUX ROBOTS SE CHERCHANT
 DANS LABYRINTHE,51 PILIERS,HEURISTIQUE
DE NAVIGATION (1),HEURISTIQUE DE DEGAGEM
ENT (C),PARCOURS D'APPRENTISSAGE UNIQUE
ET REEXECUTION,POSS CHANGMT TERRAIN

110 INK 0,16:INK 1,1:INK 2,6:INK 3,21:CL
S:CLEAR:RG=1
120 DIM C(32,25)'TABLEAU DES CASES DE L'
ECRAN SUSCEPTIBLES DE RECEVOIR UN PILIER
130 SYMBOL AFTER 32:SYMBOL 33,&0,&7C,&D6
,&6C,&38,&FE,&7C,&6C'DESSIN ROBOTS
140 'NOMBRES ALEATOIRES VARIES
150 RANDOMIZE TIME
160 CLS:H=INT(2*RND):IF H=1 THEN PXR1=IN
T(25*RND+6):PYR1=6 ELSE PXR1=6:PYR1=INT(
12*RND+6)'TIRAGE POSITION ROBOT NO 1
170 PXR2=36:PYR2=21'POSITION DEPART ROBO
T NO 2
180 PEN 2:LOCATE PXR1,PYR1:PRINT CHR$(33
):LOCATE PXR2,PYR2:PRINT CHR$(33)'AFF RO
BOTS AU DEPART
190 FOR N=0 TO 50:I=INT((33-PXR1)*RND)+P
XR1-1:J=INT((21-PYR1)/(36-PXR1)*(I-PXR1)
)+INT((5*RND-2)*(I-PXR1)/8)+PYR1+1
200 IF J>PYR1-2 THEN 220
```

```
210 J=PYR1-1:GOTO 240
220 IF J<>PYR1 THEN IF J<22 THEN 240 ELS
E J=21:GOTO 240
230 IF I<>PXR1 THEN 240 ELSE J=PYR1-1
240 PEN 1:LOCATE I,J:PRINT"="'AFF PILIER

250 C(I,J)=1:NEXT'REPRESENTATION PILIER

260 PXA=PXR1:PYA=PYR1:PPXA=PXA:PPYA=PYA'
MISE EN BOUCLE DU ROBOT NO1 (ACTIF)
270 PXP=PXR2:PYP=PYR2'MISE EN BOUCLE DU
ROBOT NO2 (PASSIF)
280 DIM PX(100),PY(100)'LISTES DES COORD
ONNEES SUCCESSIVES DES DEUX ROBOTS (INDI
CES IMPAIRS POUR ROBOT NO 1,PAIRS POUR R
OBOT NO 2)
290 SOUND 1,200,80,7:FOR AT=1 TO 2000:NE
XT'TOP DEPART
1000 REM BOUCLE DES MOUVEMENTS
1010 FOR M=1 TO 100
1020 TERR$=INKEY$:IF TERR$=" " THEN 110'
CHANGMT TERRAIN EVENTUEL
1030 PEN 1:LOCATE 31,2:PRINT M;" "'AFF C
OMPTEUR
1040 REM HEURIST NAV (1)
1050 IF PXA<>PXP THEN NUME=ABS(PYA-PYP):
DENO=ABS(PXA-PXP):IN=NUME/DENO ELSE 1080

1060 IF IN<0.5 THEN VXA=SGN(PXP-PXA):VYA
=0:GOTO 1090
1070 IF IN>2 THEN VXA=0:VYA=SGN(PYP-PYA)
:GOTO 1090
1080 VXA=SGN(PXP-PXA):VYA=SGN(PYP-PYA)
1090 TT=0'INITIALISATION DU RANG TT DES
TENTATIVES POUR TROUVER UNE POSITION LIB
RE APRES AVOIR HEURTE UN PILIER
1100 PXA=PPXA+VXA:PYA=PPYA+VYA
1110 IF PXA>31 THEN 1130 ELSE GOSUB 3000
1120 IF OBS=1 THEN 1100
1130 IF M/2=INT(M/2) THEN PER=30 ELSE PE
R=40
1140 PEN 3:LOCATE PPXA,PPYA:PRINT CHR$(3
3):PEN 2:LOCATE PXA,PYA:PRINT CHR$(33):S
OUND 1,PER,5,7:PPXA=PXA:PPYA=PYA
1150 PX(M)=PXA:PY(M)=PYA'MEMO POSITIONS
SUCCESSIVES DES DEUX ROBOTS
1160 GOSUB 4000'TEST RENCONTRE DES DEUX
ROBOTS
1170 REM INVERSION DES DEUX ROBOTS
1180 PXANOUV=PXP:PYANOUV=PYP:PXP=PXA:PYP
=PYA:PXA=PXANOUV:PYA=PYANOUV:PPXA=PXA:PP
YA=PYA
1190 FOR N=1 TO 80:NEXT:NEXT M
1200 REM REEXECUTION DU PARCOURS
```

```
1210 GOSUB 5000
1220 REM RENVOI A NOUVEAU PARCOURS
1230 FOR AT=1 TO 1000:NEXT:GOTO 110
1240 END
2000 REM ZONE DES SOUS-PROGRAMMES
3000 REM SSPROG HEURT PILIER
3010 OBS=0:TT=TT+1
3020 IF C(PXA,PYA)<>1 THEN RETURN
3030 FOR AT=1 TO 150:NEXT:SOUND 1,24,5,7
::PEN 2:LOCATE PXA,PYA:PRINT"*":FOR AT=1
 TO 50:NEXT:PEN 1:LOCATE PXA,PYA:PRINT"=
"
3040 REM HEURIST DE DEGAGMT (C):TIRAGE A
U SORT 1ERE DIRECTION DE DEGAGEMENT,PUIS
,EN CAS D'INSUCCES,EXPLORATION ORDONNEE
ES AUTRES DIRECTIONS
3050 FOR AT=1 TO 80:NEXT:IF TT=1 THEN HA
S=INT(8*RND+1)::SS=INT(2*RND):GOTO 3090
3060 IF SS=0 THEN HAS=HAS+1::IF HAS<9 TH
EN 3090 ELSE HAS=HAS-8:GOTO 3090
3070 HAS=HAS-1:IF HAS>0 THEN 3090
3080 HAS=HAS+8
3090 IF HAS=1 THEN 3100 ELSE  IF HAS=2 T
HEN 3110 ELSE IF HAS=3 THEN 3120 ELSE IF
 HAS=4 THEN 3130 ELSE IF HAS=5 THEN 3140
 ELSE IF HAS=6 THEN 3150 ELSE IF HAS=7 T
HEN 3160 ELSE 3170
3100 VXA=1:VYA=1:GOTO 3180
3110 VXA=1:VYA=0:GOTO 3180
3120 VXA=1:VYA=-1:GOTO 3180
3130 VXA=0:VYA=-1:GOTO 3180
3140 VXA=-1:VYA=-1:GOTO 3180
3150 VXA=-1:VYA=0:GOTO 3180
3160 VXA=-1:VYA=1:GOTO 3180
3170 VXA=0:VYA=1
3180 OBS=1
3190 RETURN
4000 REM SSPROG RENCONTRE DES DEUX ROBOT
S
4010 IF PXA<>PXP THEN RETURN ELSE IF PYA
<>PYP THEN RETURN
4020 PEN 1:LOCATE PXA,PYA:PRINT CHR$(33)
4030 FOR FOIS=1 TO 4:SOUND 1,27,5,7:FOR
AT=1 TO 100:NEXT:NEXT:FOR AT=1 TO 2000:N
EXT
4040 MRENC=M'MEMO NO BOUCLE DE LA RENCON
TRE
4050 PEN 3:LOCATE PXA,PYA:PRINT CHR$(33)
'AFF TRACE AU POINT DE RENCONTRE
4060 M=100'SORTIE BOUCLE DES MOUVEMENTS
4070 RETURN
5000 REM SSPROG REEXECUTION
5010 FOR FOIS=1 TO 3:SOUND 1,20,5,7:FOR
AT=1 TO 100:NEXT:NEXT:PEN 2:LOCATE PXR1,
```

```
PYR1:PRINT CHR$(33):LOCATE PXR2,PYR2:PRI
NT CHR$(33):FOR AT=1 TO 2000:NEXT'REAFF
ROBOTS AU DEPART
5020 FOR M=1 TO MRENC
5030 IF M/2=INT(M/2) THEN PER=30 ELSE PE
R=40
5040 IF M=1 THEN PEN 3:LOCATE PXR1,PYR1:
PRINT CHR$(33):GOTO 5070
5050 IF M=2 THEN PEN 3:LOCATE PXR2,PYR2:
PRINT CHR$(33):GOTO 5070
5060 PEN 3:LOCATE PX(M-2),PY(M-2):PRINT
CHR$(33)
5070 PEN 2:LOCATE PX(M),PY(M):PRINT CHR$
(33):SOUND 1,PER,5,7
5080 NEXT M
5090 FOR AT=1 TO 300:NEXT:PEN 1:LOCATE P
X(MRENC),PY(MRENC):PRINT CHR$(33):FOR FO
IS=1 TO 3:SOUND 1,18,5,7:FOR AT=1 TO 100
:NEXT:NEXT
5100 MRENC=0:RETURN
```

Commentaires techniques

- Par rapport à E16, la nouveauté réside dans l'utilisation de deux tableaux PX(100) et PY(100), préalablement dimensionnés chacun à 100 éléments à l'aide de l'instruction DIM.
- Ces tableaux prennent en charge les coordonnées successives PX(M) et PY(M) à chaque boucle de rang M.
- À l'issue du parcours, le sous-programme "rencontre des deux robots" mémorise le rang MRENC de la boucle où a eu lieu la rencontre et provoque la sortie de la boucle (M = 100).
- Puis le sous-programme réexécution procède à la reconstitution très rapide du parcours en dévidant et en affichant les positions précédemment mémorisées.

VII

LE PASSÉ AU SERVICE DU PRÉSENT — L'APPRENTISSAGE

"Le temps est un grand maître ; le malheur est qu'il tue ses élèves."

H. BERLIOZ.

Les deux robots ont commencé à prendre conscience, en E20, qu'un peu de concentration alliée à une bonne mémoire permettent de diminuer la peine et de faire le travail plus vite.

En effet, les tâtonnements du premier essai sont supprimés, le cheminement est sûr, mais cela s'arrête là : la qualité du parcours n'est pas améliorée, les écarts et les retours en arrière sont rigoureusement et stupidement restitués au second.

Or, nos parents et nos maîtres nous ont dit que l'apprentissage avait une portée bien supérieure à celle du simple fil d'Ariane : apprendre certes, mais aussi trier, éliminer, *oublier pour faire mieux et plus vite.*

Maintenant, Roméo va redevenir sérieux et solitaire, la solitude étant surtout destinée à simplifier le programme. Dans de très modestes limites, car les mécanismes de l'apprentissage sont complexes et subtils, nous allons dans les trois prochaines étapes affiner le comportement de E20, puis déboucher en E24 sur une curieuse méthode d'apprentissage dans laquelle le robot compte sur son créateur pour l'aider à mieux faire, solidarité et noblesse obligent.

ÉTAPE E21

SÉPARER LE BON GRAIN

Présentation

Le robot ne se contente plus de restituer le seul parcours qu'il a effectué. Il réalise cinq parcours complets différents sur le même terrain ; les heuristiques utilisées sont 1 et G. Le rang du parcours et le temps (c'est-à-dire le nombre de pas) sont décomptés et affichés à l'écran.

A la fin de chaque parcours, le robot compare sa performance à la meilleure des précédentes. S'il a progressé, il oublie l'ancien record et retient le nouveau. Si par contre il n'a pas fait mieux, il oublie son dernier parcours.

Lorsqu'il a terminé les cinq parcours, Roméo réexécute au pas de charge la meilleure de ses prestations.

Les différences de qualité des parcours, c'est-à-dire de temps, sont certes uniquement dues au hasard : on sait que l'heuristique de dégagement G laisse à ce dernier une place de choix.

Mais il arrive dans la vie que l'on progresse de cette façon. De nombreuses grandes inventions ou découvertes ont ainsi vu le jour car le hasard est le plus grand créateur de variété, bonne ou mauvaise, que l'on connaisse.

Il faut d'ailleurs être conscient qu'en ce domaine le hasard ne suffit pas, il faut en outre :

- que l'observateur soit psychologiquement en quête de quelque chose ;
- que l'événement soit potentiellement fécond ;
- que le chercheur le reconnaisse comme tel et se l'approprie.

Avec de la patience, simulable dans le programme par une augmentation à volonté du nombre de parcours d'essai, le résultat est surprenant : on peut arriver à un parcours final d'une grande pureté.

Programme

```
100 'ETAPE E21:ROBOT CHERCHANT SON MAITR
E DANS LABYRINTHE,70 PILIERS,HEURISTIQUE
```

```
 DE NAVIGATION (1),HEURISTIQUE DE DEGAGE
MENT (G),5 PARCOURS SUR MEME TERRAIN,CHO
IX DU MEILLEUR ET REEXECUTION
110 INK 0,19:INK 1,6:INK 2,1:INK 3,17:CL
S:CLEAR
120 DIM C(32,25)'TABLEAU DES COORDONNEES
 DES CASES DE L'ECRAN SUSCEPTIBLES DE RE
CEVOIR UN PILIER
130 DIM PXR(80),PYR(80)'DOUBLE LISTE DES
 COORDONNEES SUCCESSIVES DU ROBOT
140 DIM PXRBIS(80),PYRBIS(80)'DBLE LISTE
 DES COORD DU MEILLEUR PARCOURS
150 SYMBOL AFTER 32:SYMBOL 33,&0,&7C,&D6
,&6C,&38,&FE,&7C,&6C'DESSIN ROBOT;DESSIN
 MAITRE:CODE 250
160 MT=80:MOB=-2:RGP=1'INITIALISATION PA
RAMETRES DIVERS
170 'NOMBRES ALEATOIRES VARIES
180 RANDOMIZE TIME
190 PXR(0)=36:PYR(0)=21'POSITION ROBOT A
U DEPART
200 CLS:H=INT(2*RND):IF H=1 THEN PXM=INT
(25*RND+6):PYM=6 ELSE PXM=6:PYM=INT(12*R
ND+6)'TIRAGE POSITION MAITRE
210 PEN 1:LOCATE PXR(0),PYR(0):PRINT CHR
$(33):PEN 2:LOCATE PXM,PYM:PRINT CHR$(25
0)'AFF ROBOT ET MAITRE
220 PEN 2:LOCATE 25,1:PRINT"NO PARCOURS:
":LOCATE 25,3:PRINT"MEILL TEMPS:"
230 IF RGP>1 THEN 310
240 FOR N=1 TO 70:I=INT((33-PXM)*RND)+PX
M-1:J=INT((21-PYM)/(36-PXM)*(I-PXM))+INT
((5*RND-2)*(I-PXM)/8)+PYM+1
250 IF J>PYM-2 THEN 270
260 J=PYM-1:GOTO 290
270 IF J<>PYM THEN IF J<22 THEN 290 ELSE
 J=21:GOTO 290
280 IF I<>PXM THEN 290 ELSE J=PYM-1
290 PEN 1:LOCATE I,J:PRINT"="'AFF PILIER
300 C(I,J)=1:NEXT'REPRESENTATION PILIER
310 SOUND 1,200,80,7:FOR AT=1 TO 2000:NE
XT'TOP DEPART
320 PEN 2:LOCATE 37,1:PRINT RGP
1000 REM BOUCLE DES MOUVEMENTS
1010 FOR M=1 TO 80
1020 TERR$=INKEY$:IF TERR$=" " THEN 110'
CHANGMT DE TERRAIN
1030 PEN 2:LOCATE 31,24:PRINT M;"  "'AFF
 COMPTEUR
1040 REM HEURIST NAV (1)
1050 REM HEURISTIQUE DE DEGMT (G):SI HEU
RT A BOUCLE M=MOB,REPETITION DU MM MOUVM
T A BOUCLE MOB+1
1060 IF PXR(M-1)<>PXM THEN NUME=ABS(PYR(
```

```
M-1)-PYM):DENO=ABS(PXR(M-1)-PXM):IN=NUME
/DENO ELSE 1090
1070 IF IN<0.5 THEN VXR=SGN(PXM-PXR(M-1)
):VYR=0:GOTO 1100
1080 IF IN>2 THEN VXR=0:VYR=SGN(PYM-PYR(
M-1)):GOTO 1100
1090 VXR=SGN(PXM-PXR(M-1)):VYR=SGN(PYM-P
YR(M-1))
1100 TT=0'INITIALISATION DU RANG TT DES
TENTATIVES POUR TROUVER UNE POSITION LIB
RE APRES AVOIR HEURTE UN PILIER
1110 IF M<>MOB+1 THEN 1130
1120 PXR(M)=PXR(M-1)+VXROB:PYR(M)=PYR(M-
1)+VYROB:GOTO 1140
1130 PXR(M)=PXR(M-1)+VXR:PYR(M)=PYR(M-1)
+VYR
1140 IF PXR(M)>31 THEN 1160 ELSE GOSUB 3
000'TEST HEURT PILIER
1150 IF OBS=1 THEN 1130
1160 PEN 3:LOCATE PXR(M-1),PYR(M-1):PRIN
T CHR$(33):PEN 1:LOCATE PXR(M),PYR(M):PR
INT CHR$(33):SOUND 1,30,5,7
1170 GOSUB 4000'TEST MAITRE REJOINT
1180 FOR N=1 TO 80:NEXT:NEXT M
1190 SOUND 1,500,40,7:GOTO 4080'PARCOURS
 RATE
1200 END
2000 REM ZONE DES SOUS-PROGRAMMES
3000 REM SSPROG HEURT PILIER
3010 OBS=0:TT=TT+1
3020 IF C(PXR(M),PYR(M))<>1 THEN RETURN
3030 FOR AT=1 TO 150:NEXT:SOUND 1,24,5,7
:PEN 2:LOCATE PXR(M),PYR(M):PRINT"*":FOR
 AT=1 TO 50:NEXT:PEN 1:LOCATE PXR(M),PYR
(M):PRINT"="
3040 REM HEURISTIQUE DE DEGAGEMENT (G):T
IRAGE 1ERE DIRECTION DE DEGAGMT PUIS,EN
CAS D'INSUCCES,EXPLOITATION ORDONNEE DES
 AUTRES DIRECIONS
3050 FOR AT=1 TO 100:NEXT AT:IF TT=1 THE
N HAS=INT(8*RND+1):SS=INT(2*RND):GOTO 30
90
3060 IF SS=0 THEN HAS=HAS+1::IF HAS<9 TH
EN 3090 ELSE HAS=HAS-8:GOTO 3090
3070 HAS=HAS-1:IF HAS>0 THEN 3090
3080 HAS=HAS+8
3090 IF HAS=1 THEN 3100 ELSE  IF HAS=2 T
HEN 3110 ELSE IF HAS=3 THEN 3120 ELSE IF
 HAS=4 THEN 3130 ELSE IF HAS=5 THEN 3140
 ELSE IF HAS=6 THEN 3150 ELSE IF HAS=7 T
HEN 3160 ELSE 3170
3100 VXR=1:VYR=1:GOTO 3180
3110 VXR=1:VYR=0:GOTO 3180
3120 VXR=1:VYR=-1:GOTO 3180
```

```
3130 VXR=0:VYR=-1:GOTO 3180
3140 VXR=-1:VYR=-1:GOTO 3180
3150 VXR=-1:VYR=0:GOTO 3180
3160 VXR=-1:VYR=1:GOTO 3180
3170 VXR=0:VYR=1
3180 OBS=1:MOB=M'MEMO NO BOUCLE
3190 VXROB=VXR:VYROB=VYR'MEMO VITESSE PO
UR BOUCLE SUIVANTE MOB+1
3200 RETURN
4000 REM SSPROG MAITRE REJOINT
4010 IF PXR(M)<>PXM THEN RETURN ELSE IF
PYR(M)<>PYM THEN RETURN
4020 PEN 2:LOCATE PXR(M),PYR(M):PRINT CH
R$(33)
4030 FOR FOIS=1 TO 4:SOUND 1,27,5,7:FOR
AT=1 TO 100:NEXT:NEXT
4040 REM CHOIX MEILLEUR PARCOURS
4050 IF M>MT THEN 4080 ELSE MT=M:PEN 2:L
OCATE 37,3:PRINT MT
4060 REM MEMO MEILLEUR PARCOURS
4070 FOR M=0 TO MT:PXRBIS(M)=PXR(M):PYRB
IS(M)=PYR(M):NEXT M
4080 RGP=RGP+1:PEN 1:LOCATE PXR(0),PYR(0
):PRINT CHR$(33):PEN 2:LOCATE PXM,PYM:PR
INT CHR$(250):IF RGP<6 THEN 230
4090 PEN 1:LOCATE 38,1:PRINT"FIN"
4100 REM REEXECUTION DU MEILLEUR PARCOUR
S
4110 FOR AT=1 TO 2000:NEXT :FOR FOIS=1 T
O 4:SOUND 1,40,5,7:FOR AT=1 TO 100:NEXT:
NEXT:FOR AT=1 TO 1000:NEXT
4120 FOR M=1 TO MT:PEN 1:LOCATE PXRBIS(M
),PYRBIS(M):PRINT CHR$(33):SOUND 1,16,2,
7:FOR AT=1 TO 60:NEXT:NEXT M
4130 PEN 2:LOCATE PXM,PYM:PRINT CHR$(33)
:FOR FOIS=1 TO 4:SOUND 1,30,5,7:FOR AT=1
 TO 100:NEXT:NEXT
4140 FOR AT=1 TO 2000:NEXT:GOTO 110
```

Commentaires techniques

- On prépare, en amont de la boucle des mouvements, non seulement les deux tableaux indicés PXR(80) et PYR(80) de mémorisation des positions de chaque parcours comme en E20, mais aussi deux tableaux PXRBIS(80) et PYRBIS(80) de mémorisation des positions du *meilleur parcours*.
- On prépare également l'affichage, en haut et à droite de l'écran, du rang du parcours et du meilleur temps (nombre de pas).

- Dans la boucle des mouvements, on affiche le décompte des pas (horloge) au fur et à mesure du déroulement du parcours.
- C'est le sous-programme maître rejoint qui procède à la comparaison des performances et qui retient la meilleure en transvasant les positions PXR(M) et PYR(M) dans les tableaux PXRBIS(M) et PYRBIS(M).
- C'est également ce sous-programme qui, à l'issue du cinquième parcours, assure la réexécution en fanfare du meilleur d'entre eux.

ÉTAPE E22

NE PAS PERSISTER DANS L'ERREUR

Présentation

Roméo va franchir un second niveau dans la conception de son apprentissage.

Au lieu d'effectuer complètement chacun des cinq parcours d'essai, il surveille en permanence l'horloge et interrompt le trajet en cours dès que le temps écoulé atteint celui du meilleur parcours déjà réalisé.

Cela économise du temps et allège la mémoire mais nécessite par contre un regard permanent sur l'horloge et sur le meilleur temps.

Ces résultats et cette double opération mentale sont fidèlement traduits dans le programme ci-après.

Programme

```
100 'ETAPE E22:COMME E21,AVEC INTERRUPTI
ON D'UN PARCOURS DES QUE SON TEMPS DEPAS
SE CELUI DU MEILLEUR PRECEDENT
110 INK 0,17:INK 1,0:INK 2,6:INK 3,26:CL
S:CLEAR
120 DIM C(32,25)'TABLEAU DES COORDONNEES
 DES CASES DE L'ECRAN SUSCEPTIBLES DE RE
CEVOIR UN PILIER
130 DIM PXR(80),PYR(80)'DOUBLE LISTE DES
 COORDONNEES SUCCESSIVES DU ROBOT
140 DIM PXRBIS(80),PYRBIS(80)'DBLE LISTE
 DES COORD DU MEILLEUR PARCOURS
150 SYMBOL AFTER 32:SYMBOL 33,&0,&7C,&D6
,&6C,&38,&FE,&7C,&6C'DESSIN ROBOT;DESSIN
 MAITRE:CODE 250
160 MT=80:MOB=-2:RGP=1:MMAX=79'INITIALIS
ATION PARAMETRES DIVERS:MEILL TEMPS MT,N
O BOUCLE HEURT PILIER MOB,RANG DU PARCOU
RS RGP,POTENTIEL MAXI BOUCLES MMAX
170 'NOMBRES ALEATOIRES VARIES
180 RANDOMIZE TIME
190 PXR(0)=36:PYR(0)=21'POSITION ROBOT A
U DEPART
200 CLS:H=INT(2*RND):IF H=1 THEN PXM=INT
(25*RND+6):PYM=6 ELSE PXM=6:PYM=INT(12*R
ND+6)'TIRAGE POSITION MAITRE
210 PEN 1:LOCATE PXR(0),PYR(0):PRINT CHR
```

```
$(33):PEN 2:LOCATE PXM,PYM:PRINT CHR$(25
0)'AFF ROBOT ET MAITRE
220 PEN 2:LOCATE 25,1:PRINT"NO PARCOURS:
":LOCATE 25,3:PRINT"MEILL TEMPS:"
230 IF RGP>1 THEN 310
240 FOR N=1 TO 70:I=INT((33-PXM)*RND)+PX
M-1:J=INT((21-PYM)/(36-PXM)*(I-PXM))+INT
((5*RND-2)*(I-PXM)/8)+PYM+1
250 IF J>PYM-2 THEN 270
260 J=PYM-1:GOTO 290
270 IF J<>PYM THEN IF J<22 THEN 290 ELSE
 J=21:GOTO 290
280 IF I<>PXM THEN 290 ELSE J=PYM-1
290 PEN 1:LOCATE I,J:PRINT"="'AFF PILIER
300 C(I,J)=1:NEXT'REPRESENTATION PILIER
310 SOUND 1,200,80,7:FOR AT=1 TO 2000:NE
XT'TOP DEPART
320 PEN 2:LOCATE 37,1:PRINT RGP
1000 REM BOUCLE DES MOUVEMENTS
1010 FOR M=1 TO MMAX
1020 TERR$=INKEY$:IF TERR$=" " THEN 110'
CHANGMT DE TERRAIN
1030 PEN 2:LOCATE 31,24:PRINT M;"  "'AFF
 COMPTEUR
1040 REM HEURIST NAV (1)
1050 REM HEURISTIQUE DE DEGMT (G):SI HEU
RT A BOUCLE M=MOB,REPETITION DU MM MOUVM
T A BOUCLE MOB+1
1060 IF PXR(M-1)<>PXM THEN NUME=ABS(PYR(
M-1)-PYM):DENO=ABS(PXR(M-1)-PXM):IN=NUME
/DENO ELSE 1090
1070 IF IN<0.5 THEN VXR=SGN(PXM-PXR(M-1)
):VYR=0:GOTO 1100
1080 IF IN>2 THEN VXR=0:VYR=SGN(PYM-PYR(
M-1)):GOTO 1100
1090 VXR=SGN(PXM-PXR(M-1)):VYR=SGN(PYM-P
YR(M-1))
1100 TT=0'INITIALISATION DU RANG TT DES
TENTATIVES POUR TROUVER UNE POSITION LIB
RE APRES AVOIR HEURTE UN PILIER
1110 IF M<>MOB+1 THEN 1130
1120 PXR(M)=PXR(M-1)+VXROB:PYR(M)=PYR(M-
1)+VYROB:GOTO 1140
1130 PXR(M)=PXR(M-1)+VXR:PYR(M)=PYR(M-1)
+VYR
1140 IF PXR(M)>31 THEN 1160 ELSE GOSUB 3
000'TEST HEURT PILIER
1150 IF OBS=1 THEN 1130
1160 PEN 3:LOCATE PXR(M-1),PYR(M-1):PRIN
T CHR$(33):PEN 1:LOCATE PXR(M),PYR(M):PR
INT CHR$(33):SOUND 1,30,5,7
1170 GOSUB 4000'TEST MAITRE REJOINT
1180 FOR N=1 TO 80:NEXT:NEXT M
1190 REM INTERRUPTION PARCOURS RATE (M>M
```

```
MAX) ET RETOUR DU ROBOT AU DEPART
1200 FOR FOIS=1 TO 3:SOUND 1,40,5,7:FOR
AT=1 TO 100:NEXT:NEXT FOIS
1210 FOR M=MMAX-1 TO 0 STEP-1:SOUND 1,50
0,5,7:PEN 1:LOCATE PXR(M),PYR(M):PRINT C
HR$(33):PEN 3:LOCATE PXR(M+1),PYR(M+1):P
RINT CHR$(33):FOR AT=1 TO 70:NEXT:NEXT M
1220 REM RENVOI A UN NOUVEAU PARCOURS
1230 FOR AT=1 TO 500:NEXT:GOTO 4110
1240 END
2000 REM ZONE DES SOUS-PROGRAMMES
3000 REM SSPROG HEURT PILIER
3010 OBS=0:TT=TT+1
3020 IF C(PXR(M),PYR(M))<>1 THEN RETURN
3030 FOR AT=1 TO 150:NEXT:SOUND 1,24,5,7
:PEN 2:LOCATE PXR(M),PYR(M):PRINT"*":FOR
 AT=1 TO 50:NEXT:PEN 1:LOCATE PXR(M),PYR
(M):PRINT"="
3040 REM HEURISTIQUE DE DEGAGEMENT (G):T
IRAGE 1ERE DIRECTION DE DEGAGMT PUIS,EN
CAS D'INSUCCES,EXPLOITATION ORDONNEE DES
 AUTRES DIRECIONS
3050 FOR AT=1 TO 100:NEXT AT:IF TT=1 THE
N HAS=INT(8*RND+1):SS=INT(2*RND):GOTO 30
90
3060 IF SS=0 THEN HAS=HAS+1::IF HAS<9 TH
EN 3090 ELSE HAS=HAS-8:GOTO 3090
3070 HAS=HAS-1:IF HAS>0 THEN 3090
3080 HAS=HAS+8
3090 IF HAS=1 THEN 3100 ELSE  IF HAS=2 T
HEN 3110 ELSE IF HAS=3 THEN 3120 ELSE IF
 HAS=4 THEN 3130 ELSE IF HAS=5 THEN 3140
 ELSE IF HAS=6 THEN 3150 ELSE IF HAS=7 T
HEN 3160 ELSE 3170
3100 VXR=1:VYR=1:GOTO 3180
3110 VXR=1:VYR=0:GOTO 3180
3120 VXR=1:VYR=-1:GOTO 3180
3130 VXR=0:VYR=-1:GOTO 3180
3140 VXR=-1:VYR=-1:GOTO 3180
3150 VXR=-1:VYR=0:GOTO 3180
3160 VXR=-1:VYR=1:GOTO 3180
3170 VXR=0:VYR=1
3180 OBS=1:MOB=M'MEMO NO BOUCLE
3190 VXROB=VXR:VYROB=VYR'MEMO VITESSE PO
UR BOUCLE SUIVANTE MOB+1
3200 RETURN
4000 REM SSPROG MAITRE REJOINT
4010 IF PXR(M)<>PXM THEN RETURN ELSE IF
PYR(M)<>PYM THEN RETURN
4020 REM LE PRESENT PARCOURS EST AU MOIN
S AUSSI BON QUE LE MEILLEUR (SINON IL AU
RAIT ETE INTERROMPU)
4030 MT=M
4040 PEN 2:LOCATE PXR(M),PYR(M):PRINT CH
```

```
R$(33)
4050 FOR FOIS=1 TO 4:SOUND 1,27,5,7:FOR
AT=1 TO 100:NEXT:NEXT
4060 PEN 2:LOCATE 37,3:PRINT MT
4070 REM MEMO MEILLEUR PARCOURS
4080 FOR M=0 TO MT:PXRBIS(M)=PXR(M):PYRB
IS(M)=PYR(M):NEXT M
4090 REM ABAISSEMENT POTENTIEL BOUCLES
4100 MMAX=MT
4110 RGP=RGP+1:PEN 1:LOCATE PXR(0),PYR(0
):PRINT CHR$(33):PEN 2:LOCATE PXM,PYM:PR
INT CHR$(250):IF RGP<6 THEN 230
4120 PEN 1:LOCATE 38,1:PRINT"FIN"
4130 REM REEXECUTION DU MEILLEUR PARCOUR
S
4140 FOR AT=1 TO 2000:NEXT :FOR FOIS=1 T
O 4:SOUND 1,40,5,7:FOR AT=1 TO 100:NEXT:
NEXT:FOR AT=1 TO 1000:NEXT
4150 FOR M=1 TO MT:PEN 1:LOCATE PXRBIS(M
),PYRBIS(M):PRINT CHR$(33):SOUND 1,16,2,
7:FOR AT=1 TO 60:NEXT:NEXT M
4160 PEN 2:LOCATE PXM,PYM:PRINT CHR$(33)
:FOR FOIS=1 TO 4:SOUND 1,30,5,7:FOR AT=1
 TO 100:NEXT:NEXT
4170 FOR AT=1 TO 2000:NEXT:GOTO 110
```

Commentaires techniques

- L'interruption d'un parcours dépassant le meilleur temps MT est obtenue à l'aide du paramètre MMAX qui représente le nombre maximal de boucles admissibles et que l'on place donc en argument du début de boucle FOR. Au départ du premier parcours, MMAX est initialisé à 79. Ensuite MMAX est égalisé au meilleur temps MT.
- Lorsque le potentiel des boucles est épuisé, c'est-à-dire si le robot n'a pas rejoint son maître au meilleur temps MT (M supérieur à MMAX), le programme sort de la boucle, le robot revient sur ses pas et entreprend un nouvel essai.
- Noter que tout parcours terminé est forcément, avec cette méthode, au moins aussi bon que le meilleur précédent. Il n'est donc pas nécessaire de faire le tri comme en E21 ; le tri s'est fait par élimination en cours de trajet.
- Après chaque parcours réussi, MMAX est mis à jour : MMAX = MT.

ÉTAPE E23

DEVANCER L'ERREUR

Présentation

Dernier échelon possible dans l'économie du temps et de la mémoire, Roméo, dans chacune de ses cinq courses contre la montre, n'attend pas le moment où il a effectivement perdu pour interrompre son parcours. Il surveille, non seulement l'horloge et le meilleur temps, mais aussi son maître et la distance qui lui reste à parcourir.

Il peut en déduire à chaque pas le temps que, dans le meilleur des cas (plus d'obstacles à contourner), il pourra réaliser. Il anticipe ainsi sur ses chances et renonce dès que l'espoir de faire mieux s'envole.

L'efficacité de cette démarche se paie bien sûr par des manipulations supplémentaires d'informations.

Programme

```
100 'ETAPE E23:COMME E22,AVEC INTERRUPTI
ON ANTICIPEE D'UN PARCOURS DES QUE SON T
EMPS NE PEUT PLUS ETRE EGAL OU INFERIEUR
 A CELUI DU MEILLEUR PRECEDENT,COMPTE-TE
NU DE LA DISTANCE RESTANT A PARCOURIR
110 INK 0,17:INK 1,0:INK 2,6:INK 3,26:CL
S:CLEAR
120 DIM C(32,25)'TABLEAU DES COORDONNEES
 DES CASES DE L'ECRAN SUSCEPTIBLES DE RE
CEVOIR UN PILIER
130 DIM PXR(80),PYR(80)'DOUBLE LISTE DES
 COORDONNEES SUCCESSIVES DU ROBOT
140 DIM PXRBIS(80),PYRBIS(80)'DBLE LISTE
 DES COORD DU MEILLEUR PARCOURS
150 SYMBOL AFTER 32:SYMBOL 33,&0,&7C,&D6
,&6C,&38,&FE,&7C,&6C'DESSIN ROBOT;DESSIN
 MAITRE:CODE 250
160 MT=80:MOB=-2:RGP=1:MMAX=79'INITIALIS
ATION PARAMETRES DIVERS:MEILL TEMPS MT,N
O BOUCLE HEURT PILIER MOB,RANG DU PARCOU
RS RGP,POTENTIEL MAXI BOUCLES MMAX
170 'NOMBRES ALEATOIRES VARIES
180 RANDOMIZE TIME
190 PXR(0)=36:PYR(0)=21'POSITION ROBOT A
U DEPART
200 CLS:H=INT(2*RND):IF H=1 THEN PXM=INT
```

```
(25*RND+6):PYM=6 ELSE PXM=6:PYM=INT(12*R
ND+6)'TIRAGE POSITION MAITRE
210 PEN 1:LOCATE PXR(0),PYR(0):PRINT CHR
$(33):PEN 2:LOCATE PXM,PYM:PRINT CHR$(25
0)'AFF ROBOT ET MAITRE
220 PEN 2:LOCATE 25,1:PRINT"NO PARCOURS:
":LOCATE 25,3:PRINT"MEILL TEMPS:"
230 IF RGP>1 THEN 310
240 FOR N=1 TO 70:I=INT((33-PXM)*RND)+PX
M-1:J=INT((21-PYM)/(36-PXM)*(I-PXM))+INT
((5*RND-2)*(I-PXM)/8)+PYM+1
250 IF J>PYM-2 THEN 270
260 J=PYM-1:GOTO 290
270 IF J<>PYM THEN IF J<22 THEN 290 ELSE
 J=21:GOTO 290
280 IF I<>PXM THEN 290 ELSE J=PYM-1
290 PEN 1:LOCATE I,J:PRINT"="'AFF PILIER
300 C(I,J)=1:NEXT'REPRESENTATION PILIER
310 SOUND 1,200,80,7:FOR AT=1 TO 2000:NE
XT'TOP DEPART
320 PEN 2:LOCATE 37,1:PRINT RGP
1000 REM BOUCLE DES MOUVEMENTS
1010 FOR M=1 TO MMAX
1020 TERR$=INKEY$:IF TERR$=" " THEN 110'
CHANGMT DE TERRAIN
1030 PEN 2:LOCATE 31,24:PRINT M;"  "'AFF
 COMPTEUR
1040 REM HEURIST NAV (1)
1050 REM HEURISTIQUE DE DEGMT (G):SI HEU
RT A BOUCLE M=MOB,REPETITION DU MM MOUVM
T A BOUCLE MOB+1
1060 IF PXR(M-1)<>PXM THEN NUME=ABS(PYR(
M-1)-PYM):DENO=ABS(PXR(M-1)-PXM):IN=NUME
/DENO ELSE 1090
1070 IF IN<0.5 THEN VXR=SGN(PXM-PXR(M-1)
):VYR=0:GOTO 1100
1080 IF IN>2 THEN VXR=0:VYR=SGN(PYM-PYR(
M-1)):GOTO 1100
1090 VXR=SGN(PXM-PXR(M-1)):VYR=SGN(PYM-P
YR(M-1))
1100 TT=0'INITIALISATION DU RANG TT DES
TENTATIVES POUR TROUVER UNE POSITION LIB
RE APRES AVOIR HEURTE UN PILIER
1110 IF M<>MOB+1 THEN 1120 ELSE PXR(M)=P
XR(M-1)+VXROB:PYR(M)=PYR(M-1)+VYROB:GOTO
 1130
1120 PXR(M)=PXR(M-1)+VXR:PYR(M)=PYR(M-1)
+VYR
1130 IF PXR(M)>31 THEN 1150 ELSE GOSUB 3
000'TEST HEURT PILIER
1140 IF OBS=1 THEN 1120
1150 PEN 3:LOCATE PXR(M-1),PYR(M-1):PRIN
T CHR$(33):PEN 1:LOCATE PXR(M),PYR(M):PR
INT CHR$(33):SOUND 1,30,5,7
```

```
1160 GOSUB 4000'TEST MAITRE REJOINT
1170 REM EVALUATION DE LA POSSIBILITE DE
 TERMINER LE PARCOURS DANS LE TEMPS PREC
EDENT
1180 NA=MT-M'NA=NOMBRE MAXI ADMISSIBLE D
E MOUVEMENTS POUR EGALER LA MEILLEURE PE
RFORMANCE PRECEDENTE
1190 IF NUME>DENO THEN NN=NUME ELSE NN=D
ENO'NN=NOMBRE MINI NECESSAIRE DE MOUVEME
NTS POUR ATTEINDRE LE MAITRE
1200 IF NN<=NA THEN 1220
1210 MI=M:M=MMAX'MEMO NO BOUCLE D'INTERR
UPTION MI ET SORTIE BOUCLE
1220 IF M=MMAX THEN PEN 3:LOCATE PXR(MI)
,PYR(MI):PRINT CHR$(33)
1230 FOR N=1 TO 80:NEXT:NEXT M
1240 FOR FOIS=1 TO 3:SOUND 1,40,5,7:FOR
AT=1 TO 100:NEXT:NEXT FOIS
1250 FOR AT=1 TO 500:NEXT:GOTO 4110'RENV
OI A UN NOUVEAU PARCOURS
1260 END
2000 REM ZONE DES SOUS-PROGRAMMES
3000 REM SSPROG HEURT PILIER
3010 OBS=0:TT=TT+1
3020 IF C(PXR(M),PYR(M))<>1 THEN RETURN
3030 FOR AT=1 TO 150:NEXT:SOUND 1,24,5,7
:PEN 2:LOCATE PXR(M),PYR(M):PRINT"*":FOR
 AT=1 TO 50:NEXT:PEN 1:LOCATE PXR(M),PYR
(M):PRINT"="
3040 REM HEURISTIQUE DE DEGAGEMENT (G):T
IRAGE 1ERE DIRECTION DE DEGAGMT PUIS,EN
CAS D'INSUCCES,EXPLOITATION ORDONNEE DES
 AUTRES DIRECIONS
3050 FOR AT=1 TO 100:NEXT AT:IF TT=1 THE
N HAS=INT(8*RND+1):SS=INT(2*RND):GOTO 30
90
3060 IF SS=0 THEN HAS=HAS+1::IF HAS<9 TH
EN 3090 ELSE HAS=HAS-8:GOTO 3090
3070 HAS=HAS-1:IF HAS>0 THEN 3090
3080 HAS=HAS+8
3090 IF HAS=1 THEN 3100 ELSE  IF HAS=2 T
HEN 3110 ELSE IF HAS=3 THEN 3120 ELSE IF
 HAS=4 THEN 3130 ELSE IF HAS=5 THEN 3140
 ELSE IF HAS=6 THEN 3150 ELSE IF HAS=7 T
HEN 3160 ELSE 3170
3100 VXR=1:VYR=1:GOTO 3180
3110 VXR=1:VYR=0:GOTO 3180
3120 VXR=1:VYR=-1:GOTO 3180
3130 VXR=0:VYR=-1:GOTO 3180
3140 VXR=-1:VYR=-1:GOTO 3180
3150 VXR=-1:VYR=0:GOTO 3180
3160 VXR=-1:VYR=1:GOTO 3180
3170 VXR=0:VYR=1
3180 OBS=1:MOB=M'MEMO NO BOUCLE
```

```
3190 VXROB=VXR:VYROB=VYR'MEMO VITESSE PO
UR BOUCLE SUIVANTE MOB+1
3200 RETURN
4000 REM SSPROG MAITRE REJOINT
4010 IF PXR(M)<>PXM THEN RETURN ELSE IF
PYR(M)<>PYM THEN RETURN
4020 REM LE PRESENT PARCOURS EST AU MOIN
S AUSSI BON QUE LE MEILLEUR (SINON IL AU
RAIT ETE INTERROMPU)
4030 MT=M
4040 PEN 2:LOCATE PXR(M),PYR(M):PRINT CH
R$(33)
4050 FOR FOIS=1 TO 4:SOUND 1,27,5,7:FOR
AT=1 TO 100:NEXT:NEXT
4060 PEN 2:LOCATE 37,3:PRINT MT
4070 REM MEMO MEILLEUR PARCOURS
4080 FOR M=0 TO MT:PXRBIS(M)=PXR(M):PYRB
IS(M)=PYR(M):NEXT M
4090 REM ABAISSEMENT POTENTIEL BOUCLES
4100 MMAX=MT
4110 RGP=RGP+1:PEN 1:LOCATE PXR(0),PYR(0
):PRINT CHR$(33):PEN 2:LOCATE PXM,PYM:PR
INT CHR$(250):IF RGP<6 THEN 230
4120 PEN 1:LOCATE 38,1:PRINT"FIN"
4130 REM REEXECUTION DU MEILLEUR PARCOUR
S
4140 FOR AT=1 TO 2000:NEXT :FOR FOIS=1 T
O 4:SOUND 1,25,5,7:FOR AT=1 TO 100:NEXT:
NEXT:FOR AT=1 TO 1000:NEXT
4150 FOR M=1 TO MT:PEN 1:LOCATE PXRBIS(M
),PYRBIS(M):PRINT CHR$(33):SOUND 1,16,2,
7:FOR AT=1 TO 60:NEXT:NEXT M
4160 PEN 2:LOCATE PXM,PYM:PRINT CHR$(33)
:FOR FOIS=1 TO 4:SOUND 1,30,5,7:FOR AT=1
 TO 100:NEXT:NEXT
4170 FOR AT=1 TO 2000:NEXT:GOTO 110
```

Commentaires techniques

- C'est à chaque pas, en fin de chaque boucle après l'affichage de sa position, que le robot procède à l'estimation de ses chances d'améliorer son score.
- MT étant son meilleur temps précédent, il calcule le nombre maximal de pas admissibles NA qu'il lui faut réaliser pour ne pas dépasser son meilleur temps : NA = MT – M.
- Il calcule ensuite le nombre de pas nécessaire NN qu'il lui reste à faire dans le meilleur des cas (c'est-à-dire en supposant qu'il n'y ait plus

aucun pilier sur son chemin) pour rejoindre son maître. C'est la distance au sens où nous l'avons déjà définie dans la fonction d'évaluation de E13.

- Il ne lui reste plus qu'à comparer NN et NA : si NN est inférieur à NA, il continue ; sinon il renonce et commence un autre essai.

ÉTAPE E24

UN CURIEUX CONTRAT D'APPRENTISSAGE — DIMENSION SOCIALE DE L'INTELLIGENCE

Présentation

Roméo se dit qu'il n'a pas demandé à venir sur Terre et qu'après tout les hommes ont des devoirs envers lui.

Il se livre alors à un odieux chantage à l'amour que tout créateur porte à ses créatures, et lui demande périodiquement si les résultats qu'il obtient sont bons.

Roméo sait en effet que l'opérateur a un sens de la vue plus développé que le sien et qu'il appréhende en particulier l'ensemble des positions des piliers, alors que lui-même ne voit que son maître et doit se servir d'une canne blanche pour tâter un à un les obstacles.

Donc, aux trois premiers parcours, il utilise les heuristiques 1 et C puis il sollicite une appréciation : "C'est bien?"

Si l'opérateur lui donne une réponse positive, le robot poursuit ses trois prochains essais avec la même heuristique 1. Si le verdict est par contre négatif, il en change : au premier non, il tord l'heuristique 1 en modifiant les valeurs de IN ; il se met en somme à "loucher" en réduisant artificiellement la pente IN, ce qui dévie bien sûr les trajectoires en les aplatissant.

Au deuxième non, il louche dans l'autre sens en augmentant artificiellement la pente IN, ce qui infléchit les parcours vers le haut.

Enfin, au troisième non, Roméo pique une colère et adopte une heuristique vengeresse et imparable que le lecteur découvrira.

Puis, heureux et exténué, il s'arrête et prend cavalièrement congé de l'opérateur.

Indépendamment de ces frasques, l'opérateur peut intervenir à tout moment sur le déroulement du parcours en utilisant certaines touches du clavier : 2 pour obtenir l'heuristique 2, 4 pour l'heuristique 4, H pour l'heuristique de dégagement H. Ces deux dernières sont nouvelles et sont définies plus loin.

Ce scénario introduit en fait une idée essentielle qui ne s'est décantée que progressivement dans la dernière décennie : un programme intelligent n'est nullement obligé de fonctionner dans l'isolement. La société humaine pose même l'interactivité comme une règle quasi permanente.

Cet aspect social est maintenant largement exploité, par exemple dans

les systèmes experts où l'opérateur joue un rôle essentiel puisque c'est lui qui fournit au système les données particulières du cas à traiter (auscultation du malade, indications sur ses antécédents, etc.) et qui, en sens inverse, peut poser des questions au système sur la raison de telle ou telle déduction.

Cela implique d'ailleurs que l'opérateur soit compétent dans le domaine concerné. Il s'agit donc bien d'une mise en conjugaison de deux (ou plusieurs) intelligences, naturelles ou artificielles.

Par le même moyen d'interactivité, les connaissances générales permanentes du programme peuvent être améliorées : introduction d'un mot nouveau dans un répertoire, mise à jour d'une liste de symptômes, adjonction d'une règle nouvelle ou exclusion d'une connaissance s'avérant dépassée, etc.

Une autre notion d'ordre totalement différent apparaît dans le présent exemple. Les heuristiques insufflées au robot reviennent, par le comportement général qu'elles suscitent (les temps de réflexion, la sonorisation,...), à lui donner un certain *caractère*.

Aux réponses positives, Roméo semble content puisqu'il continue sur sa lancée.

Aux réponses négatives, il donne d'abord l'impression d'être reconnaissant puisqu'il se plie aussitôt à l'avis qu'on lui donne.

Au troisième non, par contre, son attitude donne l'image d'une exaspération mal contenue.

Ainsi l'on peut, par le jeu d'une heuristique et d'un environnement de programmation appropriés, restituer à l'opérateur, par le périphérique de sortie que constitue l'écran, une représentation externe d'un élément particulièrement subjectif et abstrait, le caractère du robot.

Les représentations qualitatives propres à l'I.A. ne sont donc pas à sens unique.

Programme

```
100 'ETAPE E24:ASSOC ROBOT-OPERATEUR;ROB
 CHERCH MTRE DS LAB,HEUR NAV ET DEGT VAR
IEES SUIVANT LES INTERVENTIONS DE L'OPER
 OU LES REPONSES QU'IL DONNE AUX QUEST P
OSEES PAR LE ROBOT
120 REM HEUR DE DEGT;PAR DEFAUT:(C);TOUC
HE H:(H)
130 INK 0,10:INK 1,1:INK 2,6:INK 3,23:GL
```

```
S:CLEAR
140 SYMBOL AFTER 32:SYMBOL 33,&0,&7C,&D6
,&6C,&38,&FE,&7C,&6C'DESSIN ROBOT;DESSIN
 MAITRE:CODE 250
150 DIM C(31,25)'TABLEAU, DES COORDONNEE
S DES CASES DE L'ECRAN SUSCEPTIBLES D'AV
OIR UN PILIER
160 'NOMBRES ALEATOIRES VARIES
170 RANDOMIZE TIME
180 RGP=0:MAL=0'RANG DU PARCOURS ET NOMB
RE DE REP NON AU ROBOT
190 CLS:H=INT(2*RND):IF H=1 THEN PXM=INT
(25*RND+6):PYM=6 ELSE PXM=6:PYM=INT(12*R
ND+6)'TIRAGE POSITION MAITRE
200 PEN 2:LOCATE PXM,PYM:PRINT CHR$(250)
'AFF MAITRE
210 IF RGP<>0 THEN 290
220 FOR N=1 TO 50:I=INT((33-PXM)*RND)+PX
M-1:J=INT((21-PYM)/(36-PXM)*(I-PXM))+INT
((5*RND-2)*(I-PXM)/8)+PYM+1
230 IF J>PYM-2 THEN 250
240 J=PYM-1:GOTO 270
250 IF J<>PYM THEN IF J<22 THEN 270 ELSE
 J=21:GOTO 270
260 IF I<>PXM THEN 270 ELSE J=PYM-1
270 PEN 1:LOCATE I,J:PRINT"=":C(I,J)=1:N
EXT N'AFF ET REPRESENTATION PILIER
280 DIM PXR(80),PYR(80)'DBLE TABLEAU DES
 COORDONNEES DES POSITIONS SUCCESSIVES D
U ROBOT
290 PXR(0)=36:PYR(0)=21:PEN 1:LOCATE PXR
(0),PYR(0):PRINT CHR$(33)'POS DEPART ET
AFF ROBOT
300 RGP=RGP+1
310 REM DEMANDE D'APPRECIATION DU RESULT
AT A L'OPERATEUR PAR LE ROBOT
320 IF RGP<4 THEN 360
330 RGP=1:PEN 1:LOCATE 31,2:PRINT"B?":FO
R AT=1 TO 100:NEXT:LOCATE 31,2:PRINT"  "
340 REP$=INKEY$:IF REP$="O" THEN 360 ELS
E IF REP$="N" THEN 350 ELSE SOUND 1,50,5
,5:GOTO 330
350 MAL=MAL+1
360 SOUND 1,200,80,7:FOR AT=1 TO 2000:NE
XT'TOP DEPART
1000 REM BOUCLE DES MOUVEMENTS
1010 FOR M=1 TO 80'80 MOUVMTS MAXI ADMIS
 POUR ATTEINDRE LE MAITRE
1020 PEN 1:LOCATE 36,24:PRINT M
1030 REM CHOIX EVENTUEL HEURIST (2),(4)
OU (H)
1040 HEUR$=INKEY$:IF HEUR$="2" THEN 1140
1050 REM HEURIST NAV (1)
1060 IF PXR(M-1)<>PXM THEN NUME=ABS(PYR(
```

```
M-1)-PYM):DENO=ABS(PXR(M-1)-PXM):IN=NUME
/DENO ELSE 1140
1070 IF MAL=0 THEN 1120 ELSE IF MAL=1 TH
EN 1110 ELSE IF MAL=2 THEN 1100 'ALTERAT
ION EVENTUELLE DE IN
1080 GOSUB 5000'CAS MAL=3 (NON A 3EME QU
ESTION DU ROBOT,HEURIST SPECIALE)
1090 GOTO 1000'FIN DU PARCOURS HEURIST S
PECIALE
1100 IN=3*IN:GOTO 1120'CAS NON A 2EME QU
EST DU ROB
1110 IN=0.3*IN'CAS NON A 1ERE QUEST DU R
OB
1120 IF IN<0.5 THEN VXR=SGN(PXM-PXR(M-1)
):VYR=0:GOTO 1150
1130 IF IN>2 THEN VXR=0:VYR=SGN(PYM-PYR(
M-1)):GOTO 1150
1140 VXR=SGN(PXM-PXR(M-1)):VYR=SGN(PYM-P
YR(M-1))
1150 REM CHOIX EVENTUEL HEURIST NAV (4)
1160 IF HEUR$="4" THEN 1170 ELSE 1180
1170 VXR=-VXR:VYR=-VYR
1180 TT=0'INITIALISATION DU RANG TT DES
TENTATIVES POUR TROUVER UNE POSITION LIB
RE APRES AVOIR HEURTE UN PILIER
1190 PXR(M)=PXR(M-1)+VXR:PYR(M)=PYR(M-1)
+VYR
1200 IF PXR(M)>31 THEN 1220 ELSE GOSUB 3
000'TEST HEURT PILIER
1210 IF OBS=1 THEN 1190
1220 PEN 3:LOCATE PXR(M-1),PYR(M-1):PRIN
T CHR$(33):PEN 1:LOCATE PXR(M),PYR(M):PR
INT CHR$(33):SOUND 1,30,5,7
1230 GOSUB 4000'TEST MAITRE REJOINT
1240 FOR N=1 TO 80:NEXT:NEXT M
1250 SOUND 1,1000,50,7:PEN 3:LOCATE PXR(
M-1),PYR(M-1):PRINT CHR$(33):FOR AT=1 TO
 300:NEXT:GOTO 200'PARCOURS RATE
1260 END
2000 REM ZONE DES SOUS-PROGRAMMES
3000 REM SSPROG HEURT PILIER
3010 OBS=0:TT=TT+1
3020 IF C(PXR(M),PYR(M))<>1 THEN RETURN
3030 FOR AT=1 TO 150:NEXT:SOUND 1,24,5,7
:PEN 2:LOCATE PXR(M),PYR(M):PRINT"*":FOR
 AT=1 TO 50:NEXT:PEN 1:LOCATE PXR(M),PYR
(M):PRINT"="
3040 REM HEURISTIQUE DE DEGAGEMENT (C):T
IRAGE 1ERE DIRECTION DE DEGAGMT PUIS,EN
CAS D'INSUCCES,EXPLOITATION ORDONNEE DES
 AUTRES DIRECIONS
3050 FOR AT=1 TO 100:NEXT AT:IF TT=1 THE
N HAS=INT(8*RND+1):SS=INT(2*RND):GOTO 30
90
```

```
3060 IF SS=0 THEN HAS=HAS+1::IF HAS<9 TH
EN 3090 ELSE HAS=HAS-8:GOTO 3090
3070 HAS=HAS-1:IF HAS>0 THEN 3090
3080 HAS=HAS+8
3090 IF HAS=1 THEN 3100 ELSE  IF HAS=2 T
HEN 3110 ELSE IF HAS=3 THEN 3120 ELSE IF
 HAS=4 THEN 3130 ELSE IF HAS=5 THEN 3140
 ELSE IF HAS=6 THEN 3150 ELSE IF HAS=7 T
HEN 3160 ELSE 3170
3100 VXR=1:VYR=1:GOTO 3180
3110 VXR=1:VYR=0:GOTO 3180
3120 VXR=1:VYR=-1:GOTO 3180
3130 VXR=0:VYR=-1:GOTO 3180
3140 VXR=-1:VYR=-1:GOTO 3180
3150 VXR=-1:VYR=0:GOTO 3180
3160 VXR=-1:VYR=1:GOTO 3180
3170 VXR=0:VYR=1
3180 REM CHOIX EVENTUEL HEURIST DEGMT (H
)
3190 IF HEUR$="H" THEN 3200 ELSE 3210
3200 VXR=2*VXR:VYR=2*VYR
3210 OBS=1
3220 RETURN
4000 REM SSPROG MAITRE REJOINT
4010 IF PXR(M)<>PXM THEN RETURN ELSE IF
PYR(M)<>PYM THEN RETURN
4020 FOR FOIS=1 TO 4:SOUND 1,27,5,7:FOR
AT=1 TO 100:NEXT:NEXT
4025 COL=1
4030 FOR AT=1 TO 800:NEXT:IF COL=0 THEN
200
4040 FOR FOIS =1 TO 4:PAPER 2:CLS:SOUND
1,90,20,7:FOR AT=1 TO 200:NEXT:PAPER 0:C
LS:NEXT FOIS
4050 CLS:PEN 2:LOCATE 11,11:PRINT"ASSEZ
POUR AUJOURD'HUI..."
4060 END
5000 REM HEURISTIQUE SPECIALE COLERE ROB
OT
5010 PEN 2:LOCATE 36,21:PRINT CHR$(33):F
OR FOIS=1 TO 4:SOUND 1,100,5,7:FOR AT=1
TO 100:NEXT:NEXT FOIS
5020 FOR N=0 TO 32:PEN 2:LOCATE 36-N,21:
PRINT CHR$(33):PEN 3:LOCATE 36-N,21:PRIN
T CHR$(33):NEXT
5030 FOR N=1 TO 18:PEN 2:LOCATE 4,21-N:S
OUND 1,70,5,7:PEN 3:LOCATE 4,21-N:PRINT
CHR$(33):NEXT
5040 PXR(0)=4:PYR(0)=2:PEN 1:LOCATE PXR(
0),PYR(0):PRINT CHR$(33)
5050 MAL=0:COL=1'REINIT MAL ET SIGNALMT
COLERE
5060 RETURN
```

Commentaires techniques

- Les heuristiques normales du robot, c'est-à-dire sans intervention spéciale de l'opérateur, sont comme nous l'avons vu : 1 pour la navigation et C pour les dégagements.
- Les changements d'heuristiques sont provoqués au clavier :

  Tous les trois parcours, par la réponse O ou N à la question "B ?" (bien ?) posée par le robot ;

  À un moment quelconque d'un parcours, par des changements ponctuels à l'aide des touches 2, 4 ou H.
- La question "B ?" est programmée en amont de la boucle des mouvements, tous les trois parcours, à l'aide du paramètre RGP (rang du parcours) préalablement initialisé à 0, puis réinitialisé à 1 lors de chaque réponse.
- Chaque réponse négative (touche N) se traduit par l'incrémentation du paramètre MAL, initialisé à 0, qui sert à aiguiller le programme sur l'heuristique adéquate : 1 si MAL = 0, 1 "réduite" si MAL = 1, 1 "dilatée" si MAL = 2 et heuristique spéciale-colère si MAL = 3.

  Si MAL = 0, IN joue son rôle habituel ;

  Si MAL = 1, IN est réduit : IN = 0,3 × IN ;

  Si MAL = 2, IN est dilaté : IN = 3 × IN ;

  Si MAL = 3, renvoi au sous-programme heuristique spéciale-colère.
- Les interventions au clavier pour modifier le comportement du robot en cours de route (touches 2, 4 et H) sont programmées au début de la boucle des mouvements à l'aide du paramètre réponse HEUR$ qui prend alors la valeur "2", "4" ou "H".
- Si HEUR$ = "2", heuristique 2.
- Si HEUR$ = "4", heuristique 4, qui constitue le négatif de l'heuristique 1 et correspond donc à une fuite du robot dans la direction opposée à son maître ; il suffit donc de prendre les valeurs — VXR et — VYR de l'heuristique 1.
- Si HEUR$ = "H", heuristique H qui consiste pour le robot à se dégager en sautant une case suivant la direction de l'heuristique C ; il suffit donc de doubler les composantes VXR et VYR de l'heuristique C : c'est ce que fait dans ce cas le sous-programme heurt pilier.
- Dans les autres cas, heuristiques 1 et C.

- Sous-programme heuristique spéciale : le robot emprunte un chemin détourné contournant l'ensemble des piliers. D'abord, un parcours horizontal jusqu'au bord gauche de l'écran, puis une remontée vers l'angle supérieur gauche. Le sous-programme se termine alors et le robot achève normalement son parcours avec les heuristiques 1 et C. Un drapeau COL = 1 (colère) annonce la fin des opérations ; c'est le sous-programme maître rejoint qui exploite ce signal et donne congé à l'opérateur sous la forme : ''assez pour aujourd'hui !''

VIII

GÉNÉRALISATION DE L'UTILISATION DES FONCTIONS LOGIQUES

"L'œuvre d'art, c'est une idée qu'on exagère."
A. GIDE.

En E18, la fonction logique C$(I, J) nous a permis de montrer comment on peut diversifier pour l'ordinateur les éléments de l'univers à représenter : piliers, Gluons et robot passif.

Nous allons, à l'aide des deux exemples suivants, illustrer le degré de puissance et de généralité de ce concept. Puissance en E25, généralité en E26 et aussi, dans les deux cas, simplicité. Il fait en somme partie de ces choses dont Audiberti disait qu'elles pouvaient être géniales sans être pour autant idiotes.

La fonction logique utilisée dans les deux exemples prendra, non plus des valeurs chaînes comme en E18, mais des valeurs numériques. Ces deux formes, interchangeables et équivalentes en puissance, sont toutes deux accessibles en BASIC.

Par contre, le BASIC ne permet ordinairement pas d'utiliser des arguments (I, J) autres que numériques et nous prive par exemple du superbe mode de représentation suivant :

> si I est un objet et si J est son propriétaire,
> si la fonction logique QAL(I, J) prend comme valeur une qualité de l'objet considéré,

on pourrait alors écrire dans une base de données :

> QAL(POMME, JEAN) = ROUGE ou
> QAL(CHAMP, JACQUES) = FERTILE

Mais cela n'est pas très gênant car l'on peut toujours codifier les objets et les propriétaires par deux séries de nombres I et J.

ÉTAPE E25

LA PUISSANCE SIMPLIFICATRICE

Présentation

Reprenons l'exemple E13, avec le robot, les piliers et les heuristiques 1 et F, cette dernière interdisant notamment, comme déjà vu, les retours en arrière.

Nous allons représenter cette fois *tous* les éléments de l'univers du robot par la fonction logique C(I, J), prenant des valeurs diverses selon les éléments présents dans la case de coordonnées I, J :

case vide	C(I, J) = 0
pilier	C(I, J) = 1
maître	C(I, J) = 2
trace	C(I, J) = RG + 2 (RG : rang du parcours).

Les traces des pas précédents sont donc intégrées dans l'univers du robot qui pourra ainsi tenir compte facilement de leur présence sans recourir aux artifices de E13 (VVXR et VVYR).

On a donné aux cases concernées la valeur variable RG + 2, de manière à offrir une option à l'opérateur : il peut désirer que seuls les retours sur les traces du parcours en cours soient prohibés (cas de E13), ou que *toutes les traces*, y compris celles des parcours précédents, lui soient interdites.

Pour savoir s'il y a ou non une trace dans la case où le robot veut aller, il suffit de tester la valeur de la fonction C(PXR, PYR). Si C vaut RG + 2, on se trouve sur une trace du parcours en cours ; si C est égale ou supérieure à 3, on est sur une trace de l'un quelconque des parcours.

C'est l'un ou l'autre de ces deux tests qui est mis en œuvre dans le sous-programme spécialisé suivant l'option choisie par l'opérateur.

Outre cette possibilité d'offrir de manière simple une option, l'extension de la fonction logique aux traces allège le sous-programme et le programme principal (suppression de la mémorisation de VVXR et VVYR).

Programme

```
100 'ETAPE E25:CMME E13,EN UTIL FONCT LO
GIQUE C(I,J) POUR REPRES LA TOTALITE DES
```

```
 ELEMENTS DE L'UNIVERS DU ROB:PILS,MTRE,
TRACES;L'HEUR DE DEGT (F) EST LEGERMT MO
DIF:INTERDICT RET SUR TTE TRACE PRECDTE
ET NON PLUS SEULMT SUR LA DERNIERE
110 INK 0,22:INK 1,1:INK 2,6:INK 3,14:CL
S:CLEAR
120 REM OPTION SUR TRACES:INTERDIC RETOU
R SUR TRACES DES PARCOURS PRECEDENTS OU
SEULEMENT SUR CELLES DU PARCOURS EN COUR
S?
130 CLS:PEN 1:LOCATE 6,9:INPUT "TRACES D
ES PARCOURS PRECEDENTS
                                INTERDITES
(O POUR OUI,N POUR NON)";R$
140 IF (R$<>"O" AND R$<>"N") THEN 130
150 DIM C(40,25)'TABLEAU DES COORDONNEES
 DES CASES DE L'ECRAN SUSCEPTIBLES DE RE
CEVOIR UN ELEMENT DE L'UNIVERS DU ROBOT
160 REM VALEURS DE LA FONCTION LOGIQUE C
(I,J) POUR LA CASE DE COORDONNEES I,J;CA
SE VIDE:C=0;PILIER:C=1;MAITRE:C=2;TRACE:
C=RG+2
170 SYMBOL AFTER 32:SYMBOL 33,&0,&7C,&D6
,&6C,&38,&FE,&7C,&6C'DESSIN ROBOT;DESSIN
 MAITRE:CODE 250
180 'NOMBRES ALEATOIRES VARIES
190 RANDOMIZE TIME
200 CLS:RG=1:H=INT(2*RND):IF H=1 THEN PX
M=INT(25*RND+6):PYM=6 ELSE PXM=6:PYM=INT
(12*RND+6)'TIRAGE POSITION MAITRE
210 PEN 2:LOCATE PXM,PYM:PRINT CHR$(250)
:C(PXM,PYM)=2'AFF ET REPRESENTATION MAIT
REMAITRE
220 PXR=36:PYR=21:PPXR=PXR:PPYR=PYR:PEN
1:LOCATE PXR,PYR:PRINT CHR$(33)'AFF ROBO
T DEPART ET MISE EN RES POUR TRACE
230 IF RG>1 THEN 1000
240 REM TIRAGE 31 PILIERS
250 FOR N=0 TO 30:TERR$=INKEY$:IF TERR$<
>" " THEN 260 ELSE 110
260 I=INT((33-PXM)*RND)+PXM-1:J=INT((21-
PYM)/(36-PXM)*(I-PXM))+INT((5*RND-2)*(I-
PXM)/8)+PYM+1
270 IF J>PYM-2 THEN 290
280 J=PYM-1:GOTO 310
290 IF J<>PYM THEN IF J<22 THEN 310 ELSE
 J=21:GOTO 310
300 IF I<>PXM THEN 310 ELSE J=PYM-1
310 PEN 1:LOCATE I,J:PRINT"=":C(I,J)=1:N
EXT'AFF ET REPRESENTATION PILIER
320 SOUND 1,200,80,7:FOR AT=1 TO 2000:NE
XT'TOP DEPART
1000 REM BOUCLE DES MOUVEMENTS
1010 FOR M=1 TO 50'50 MOUVMTS MAXI ADMIS
```

```
 POUR ATTEINDRE MAITRE
1020 REM HEURIST NAV (1)
1030 IF PXR<>PXM THEN NUME=ABS(PYR-PYM):
DENO=ABS(PXR-PXM):IN=NUME/DENO ELSE 1060
1040 IF IN<0.5 THEN VXR=SGN(PXM-PXR):VYR
=0:GOTO 1070
1050 IF IN>2 THEN VXR=0:VYR=SGN(PYM-PYR)
:GOTO 1070
1060 VXR=SGN(PXM-PXR):VYR=SGN(PYM-PYR)
1070 TT=0'INITIALISATION DU RANG TT DES
TENTATIVES POUR TROUVER UNE POSITION LIB
RE APRES AVOIR HEURTE UN PILIER
1080 PXR=PPXR+VXR:PYR=PPYR+VYR
1090 GOSUB 3000'TEST HEURT PILIER
1100 IF OBS=0 THEN 1110 ELSE 1080'RECALC
UL POS
1110 IF RET=0 THEN 1120 ELSE 1080
1120 IF M<>MPI THEN 1250
1130 REM HEURIST DE DEGT (F):ESSAI AUTRE
 SENS D'EXPLORATION
1140 IF ESS=1 THEN 1180 ELSE PEN 2:LOCAT
E PXR,PYR:PRINT"?"::PXR1=PXR:PYR1=PYR:VX
R1=VXR:VYR1=VYR'AFF ET MEMO 1ERE POS
1150 FOR AT=1 TO 100:NEXT:SSS=SS:IF SSS=
1 THEN SS=0 ELSE SS=1'INVERSION SENS DE
ROTATION
1160 TT=2:ESS=1:D=D1:GOSUB 3250
1170 GOTO 1080'RECALCUL POS
1180 ESS=0:MPI=0'REINIT ESS ET MPI
1190 PEN 2:LOCATE PXR,PYR:PRINT"?":FOR A
T=1 TO 100:NEXT'AFF 2EME POS
1200 REM HEUR DE DEGT (F):CHOIX MEILLEUR
E POSITION,PLUS COURTE DES 2 DISTANCES D
IST1 ET DIST2 AU MAITRE
1210 HOR1=ABS(PXM-PXR1):VERT1=ABS(PYM-PY
R1)::IF HOR1>VERT1 THEN DIST1=HOR1 ELSE
DIST1=VERT1
1220 HOR2=ABS(PXM-PXR):VERT2=ABS(PYM-PYR
):IF HOR2>VERT2 THEN DIST2=HOR2 ELSE DIS
T2=VERT2
1230 IF DIST2<=DIST1 THEN 1240 ELSE LOCA
TE PXR,PYR:PRINT" ":PXR=PXR1:PYR=PYR1:VX
R=VXR1:VYR=VYR1:GOTO 1250'1ERE POS CHOIS
IE
1240 LOCATE PXR1,PYR1:PRINT" "'2EME POS
CHOISIE
1250 PEN 3:LOCATE PPXR,PPYR:PRINT CHR$(3
3):C(PPXR,PPYR)=RG+2:PEN 1:LOCATE PXR,PY
R:PRINT CHR$(33):SOUND 1,30,5,7:PPXR=PXR
:PPYR=PYR'AFF ET REPRESENTATION TRACE
1260 PEN 1:LOCATE PXR,PYR:PRINT CHR$(33)
:SOUND 1,30,5,7:PPXR=PXR:PPYR=PYR
1270 GOSUB 4000'TEST MAITRE REJOINT
1280 NEXT M
```

```
1290 SOUND 1,500,40,7:GOTO 4030'PARCOURS
 RATE
1300 END
2000 REM ZONE DES SOUS-PROGRAMMES
3000 REM SSPROG HEURT PILIER ET/OU RETOU
R SUR TRACE PRECEDENTE
3010 OBS=0:RET=0
3020 IF C(PXR,PYR)=1 THEN 3030 ELSE 3040
'TEST PILIER
3030 SOUND 1,50,5,7:PEN 2:LOCATE PXR,PYR
:PRINT"*":FOR AT=1 TO 30:NEXT:PEN 1:LOCA
TE PXR,PYR:PRINT"=":OBS=1:GOTO 3080
3040 IF R$="O" THEN 3060'TEST RETOUR SUR
 TRACE PRECEDENTE
3050 IF C(PXR,PYR)=RG+2 THEN 3070 ELSE R
ETURN
3060 IF C(PXR,PYR)>=3 THEN 3070 ELSE RET
URN
3070 IF TT>7 THEN RETURN ELSE SOUND 1,20
,5,7:PEN 2:LOCATE PXR,PYR:PRINT"*":FOR A
T=1 TO 30:NEXT:PEN 3:LOCATE PXR,PYR:PRIN
T CHR$(33):RET=1
3080 FOR AT=1 TO 50:NEXT:TT=TT+1:MPI=M'M
EMO NO BOUCLE
3090 REM ESSAI NOUVELLE DIRECTION
3100 REM HEURISTIQUE DE DEGAGEMENT (F):E
XPLORATION ORDONNEE DES  DIRECTIONS EN P
ARTANT DU PILIER HEURTE
3110 FOR AT=1 TO 50:NEXT AT:IF TT>1 THEN
 3250'CAS 2EME PASSAGE ET SUIVANTS
3120 IF VXR=-1 THEN 3150'CAS 1ER PASSAGE
,DETERMINATION DE LA DIRECTION D1 DU 1ER
 HEURT
3130 IF VXR=0 THEN 3180
3140 IF VXR =1 THEN 3200
3150 IF VYR=-1 THEN D1=5:GOTO 3230
3160 IF VYR=0 THEN D1=6:GOTO 3230
3170 IF VYR=1 THEN D1=7:GOTO 3230
3180 IF VYR=-1 THEN D1=4:GOTO 3230
3190 IF VYR=1 THEN D1=8:GOTO 3230
3200 IF VYR=-1 THEN D1=3:GOTO 3230
3210 IF VYR=0 THEN D1=2:GOTO 3230
3220 IF VYR=1 THEN D1=1
3230 D=D1'DIRECTION D DEBUT DE L'EXPLORA
TION
3240 SS=INT(2*RND)'TIRAGE SENS DE L'EXPL
ORATION
3250 REM EXPLORATION POSITION SUIVANTE
3260 IF SS=0 THEN D=D+1:IF D<9 THEN 3280
 ELSE D=D-8:GOTO 3280
3270 D=D-1:IF D>0 THEN 3280 ELSE D=D+8
3280 IF D=1 THEN 3290 ELSE IF D=2 THEN 3
300 ELSE IF D=3 THEN 3310 ELSE IF D=4 TH
EN 3320 ELSE IF D=5 THEN 3330 ELSE IF D=
```

```
6 THEN 3340 ELSE IF D=7 THEN 3350 ELSE 3
360
3290 VXR=1:VYR=1:GOTO 3370
3300 VXR=1:VYR=0:GOTO 3370
3310 VXR=1:VYR=-1:GOTO 3370
3320 VXR=0:VYR=-1:GOTO 3370
3330 VXR=-1:VYR=-1:GOTO 3370
3340 VXR=-1:VYR=0:GOTO 3370
3350 VXR=-1:VYR=1:GOTO 3370
3360 VXR=0:VYR=1
3370 RETURN
4000 REM SSPROG MAITRE REJOINT
4010 IF C(PXR,PYR)<>2 THEN RETURN
4020 FOR FOIS=1 TO 4:SOUND 1,27,5,7:FOR
AT=1 TO 100:NEXT:NEXT
4030 FOR AT=1 TO 800:NEXT:RG=RG+1:IF RG<
6 THEN PEN 3:LOCATE PPXR,PPYR:PRINT CHR$
(33):GOTO 210 ELSE 110
```

Commentaires techniques

- L'option sur les traces à interdire est proposée en tête du programme à l'aide de l'instruction INPUT, à laquelle la réponse "O" ou "N" est donnée au clavier (paramètre R$).
- La trace de chaque pas est affichée en temps voulu dans la boucle des mouvements, puis représentée aussitôt par C(PPXR, PPYR) = RG + 2.
- Comme indiqué précédemment, le sous-programme heurt pilier ou retour en arrière teste la valeur de la fonction C(PXR, PYR). Les valeurs qui l'intéressent sont 1, RG + 2 ou égales ou supérieures à 3.
- Il en est de même, bien entendu, pour le sous-programme maître rejoint. La valeur qui l'intéresse est 2.

ÉTAPE E26

CINQ SENS POUR UN DANGEREUX ITINÉRAIRE

Présentation

Pour aller au bout de leur logique, les fonctions bien nommées vont fleurir dans un labyrinthe fourre-tout spécialement conçu.

Le robot va y trouver en effet :

- des cases vides où il pourra poser ses pieds : C(I, J) = 0 ;
- des piliers : C(I, J) = 1 ;
- son maître : C(I, J) = 2 ;
- des victuailles : C(I, J) = 3 ;
- des trappes ; C(I, J) = 5 ;
- des sonneries susceptibles de lui redonner du moral : C(I, J) = 6 ;
- enfin, une zone odorante au voisinage des victuailles : C(I, J) = 4.

On aurait pu en rajouter, avec des peaux de banane, des toboggans, des abreuvoirs, etc., et l'on pourrait décrire ainsi à l'ordinateur un monde très riche dans lequel chaque case d'écran recélerait un élément différent, la seule limite à cette diversification étant la résolution de l'écran.

Les positions de tous les éléments sont tirées au hasard et affichées à l'écran.

Roméo tente de rejoindre son maître à travers ce labyrinthe en utilisant les heuristiques 1 et F.

En plus des éléments externes énumérés plus haut, il aura aussi affaire à ses propres traces puisqu'il lui est interdit, dans l'heuristique F, de revenir dessus. Pour ces traces, la fonction logique prend la valeur RG + 6, pour ne tenir compte que de celles du trajet en cours (voir E25).

Le robot prend le départ avec un potentiel vital de 25 points, potentiel qui ne doit jamais devenir négatif sous peine d'interruption du parcours, et qui est en permanence affiché à l'écran.

Des sous-programmes spécialisés règlent comme d'habitude les conséquences de toutes les rencontres, heureuses ou non, que le robot est susceptible de faire au cours de son trajet, en testant chaque fois la valeur de la fonction logique à l'endroit considéré. Les règles sont les suivantes :

- chaque pas lui coûte 1 point de vitalité ;
- chaque passage sur une trappe lui retire 4 points ;
- chaque absorption de victuailles lui redonne 20 points ;
- chaque parcours réussi (maître rejoint) lui donne 15 points ;
- chaque parcours raté (potentiel vital devenu négatif) constitue un échec, mais nécessite une nouvelle dotation de 25 points pour lui permettre de commencer un nouveau parcours ;
- chaque sonnerie dont il réussit à deviner la provenance exacte lui rapporte 10 points.

Le robot effectue ainsi sept parcours, sources de péripéties assez complexes qui permettent de voir les divers effets provoqués par chaque événement.

Outre la diversité de ces effets accumulés, l'intérêt du présent exemple consiste dans la simulation de *nouveaux sens de perception* du robot.

Rappelons que le toucher intervient lorsque l'élément concerné est détecté au moment où le robot arrive dessus : c'est le cas des piliers, des trappes, des victuailles et du maître rejoint.

La vue permet quant à elle de détecter un élément éloigné, et d'en apprécier la direction et dans certains cas la distance : c'est le cas du maître pour les besoins de l'heuristique de navigation.

Cette fois, Roméo sera doté de trois sens supplémentaires :

- *l'ouïe*, qui lui permet de détecter l'activité d'une sonnerie et de savoir dans quelle direction approximative elle se situe : il doit ensuite en rechercher l'emplacement exact par déductions successives, son maître (le programme) lui indiquant chaque fois s'il a visé trop à gauche ou trop à droite ;
- *l'odorat*, qui lui permet de savoir s'il se trouve à proximité immédiate d'un stock de victuailles, c'est-à-dire dans l'une des huit cases qui entourent ce stock. Comme dans la réalité, ce sens est donc, d'une part de portée limitée, et d'autre part non directif : le robot ignore de quel côté arrive l'odeur, et il doit trouver ensuite la bonne case par exploration tactile ;
- *la faim*, simulée par un potentiel vital suffisamment faible pour qu'il se détourne de son chemin lorsqu'une odeur alléchante se manifeste.

Toutes ces perceptions doivent être bien entendu simulées et traitées

d'une manière correcte dans le programme. Il en résulte d'ailleurs que ce programme est assez important puisqu'il occupe plus de quarante kilo octets de mémoire vive.

Programme

```
100 'ETAPE E26:GENERALISTION UTILIS FONC
T LOGIQUE C(I,J) POUR REPRES LES ELEMENT
S D'UN UNIVERS COMPLEXE DU ROB:PIL,MTRE,
VICTUAILLES,ODEUR VICTUAILLES,TRAPPES,SO
NNERIES,TRACES;HEUR NAV (1),HEUR DEGT (F
)
110 INK 0,22:INK 1,1:INK 2,6:INK 3,14:CL
S:CLEAR
120 DIM C(40,25)'TABLEAU DES COORDONNEES
 DES CASES DE L'ECRAN SUSCEPTIBLES DE RE
CEVOIR UN ELEMENT DE L'UNIVERS DU ROBOT
130 REM VALEURS DE LA FONCTION LOGIQUE C
(I,J) POUR LA CASE DE COORDONNEES I,J;CA
SE VIDE:C=0;PILIER:C=1;MAITRE:C=2;VICTUA
ILLES:C=3;ENTOURAGE VICT:C=4;TRAPPE:C=5;
SONNERIE:C=6;TRACE:C=RG+6 (POUR NE TENIR
 COMPTE QUE DES TRACES DU PARCOURS EN CO
URS)
140 SYMBOL AFTER 32:SYMBOL 33,&0,&7C,&D6
,&6C,&38,&FE,&7C,&6C'DESSIN ROBOT;DESSIN
 MAITRE:CODE 250;DESSIN TRAPPE:CODE 127
150 'NOMBRES ALEATOIRES VARIES
160 RANDOMIZE TIME
170 CLS:RG=1:MVI=50:VI=25:GOSUB 8000'INI
TIALISATION RANG PARCOURS RG,NO BCLE DE
RAVITAILLEMT MVI ET RESERVE VITALE VIDU
ROBOT,ET AFF VI
180 PXM=8:PYM=INT(16*RND)+5'TIRAGE POS M
AITRE
190 PEN 2:LOCATE PXM,PYM:PRINT CHR$(250)
:C(PXM,PYM)=2'AFF ET REPRESENTATION MAIT
RE
200 REM TIRAGE 52 PILIERS,5 STOCKS DE VI
CTUAILLES ET 10 TRAPPES
210 C(7,PYM)=1:PEN 1:LOCATE 7,PYM:PRINT"
=":C(8,PYM+1)=1:LOCATE 8,PYM+1:PRINT"="
220 FOR N=1 TO 50:I=4*INT(5*RND)+9:J=INT
((19-PYM)/21*(I-8))+INT((5*RND-2)*(I-8)/
10)+PYM+1
230 IF J<23 THEN C(I,J)=1:PEN 1:LOCATE I
,J:PRINT"=" ELSE J=22:GOTO 230'PILIERS
240 IF INT(N/5)<>N/5 THEN 250 ELSE IF C(
I-1,J+1)<>1 THEN C(I,J+1)=5:PEN 2:LOCATE
 I,J+1:PRINT CHR$(127)'TRAPPES
250 NEXT N
```

```
260 FOR N=1 TO 5:I=4*N+7:J=INT((19-PYM)/
21*(I-8))+PYM+1:IF J<23 THEN 270 ELSE J=
22
270 FOR A=I-1 TO I+1:FOR B=J-1 TO J+1:C(
A,B)=4:PEN 3:LOCATE A,B:PRINT"-":NEXT B:
NEXT A'ENTOURAGE VICTUAILLES
280 C(I,J)=3:PEN 2:LOCATE I,J:PRINT"^":N
EXT N'VICTUAILLES
290 PXS1=INT(31*RND)+2:PYS1=INT(2*RND)+3
:PXS2=INT(31*RND)+2:PYS2=INT(2*RND)+23:C
(PXS1,PYS1)=6:C(PXS2,PYS2)=6:PEN 2:LOCAT
E PXS1,PYS1:PRINT"S":LOCATE PXS2,PYS2:PR
INT"S"'SONNERIES
300 PXR=29:PYR=19:PPXR=PXR:PPYR=PYR:PEN
1:LOCATE PXR,PYR:PRINT CHR$(33)'ROBOT AU
 DEPART
310 SOUND 1,200,80,7:FOR AT=1 TO 2000:NE
XT'TOP DEPART
1000 REM BOUCLE DES MOUVEMENTS
1010 FOR M=1 TO 50
1020 GOSUB 7000'TEST SONNERIE
1030 IF M<>MVI THEN 1050 ELSE GOSUB 5000
'CONSOMMATION VICTUAILLES
1040 GOTO 1340
1050 REM HEURIST NAV (1)
1060 IF PXR<>PXM THEN NUME=ABS(PYR-PYM):
DENO=ABS(PXR-PXM):IN=NUME/DENO ELSE 1090

1070 IF IN<0.5 THEN VXR=SGN(PXM-PXR):VYR
=0:GOTO 1100
1080 IF IN>2 THEN VXR=0:VYR=SGN(PYM-PYR)
:GOTO 1100
1090 VXR=SGN(PXM-PXR):VYR=SGN(PYM-PYR)
1100 TT=0'INITIALISATION DU RANG TT DES
TENTATIVES POUR TROUVER UNE POSITION LIB
RE APRES AVOIR HEURTE UN PILIER
1110 PXR=PPXR+VXR:PYR=PPYR+VYR
1120 GOSUB 3000'TEST HEURT PILIER
1130 IF OBS=0 THEN 1140 ELSE 1110'RECALC
UL POS
1140 IF RET=0 THEN 1150 ELSE 1110
1150 IF M<>MPI THEN 1300
1160 REM HEURIST DE DEGT (F):ESSAI AUTRE
 SENS D'EXPLORATION
1170 IF ESS=1 THEN 1210 ELSE PEN 2:LOCAT
E PXR,PYR:PRINT"?"::PXR1=PXR:PYR1=PYR:VX
R1=VXR:VYR1=VYR'AFF ET MEMO 1ERE POS
1180 FOR AT=1 TO 50:NEXT:SSS=SS:IF SSS=1
 THEN SS=0 ELSE SS=1'INVERSION SENS DE R
OTATION
1190 TT=2:ESS=1:D=D1:GOSUB 3230
1200 GOTO 1110'RECALCUL POS
1210 ESS=0:MPI=0'REINIT ESS ET MPI
1220 PEN 2:LOCATE PXR,PYR:PRINT"?":FOR A
```

```
T=1 TO 100:NEXT'AFF 2EME POS
1230 REM HEUR DE DEGT (F):CHOIX MEILLEUR
E POSITION,PLUS COURTE DES 2 DISTANCES D
IST1 ET DIST2 AU MAITRE
1240 HOR1=ABS(PXM-PXR1):VERT1=ABS(PYM-PY
R1)::IF HOR1>VERT1 THEN DIST1=HOR1 ELSE
DIST1=VERT1
1250 HOR2=ABS(PXM-PXR):VERT2=ABS(PYM-PYR
):IF HOR2>VERT2 THEN DIST2=HOR2 ELSE DIS
T2=VERT2
1260 IF DIST2<=DIST1 THEN 1290 ELSE 1270
1270 IF C(PXR,PYR)<>5 THEN IF C(PXR,PYR)
<>4 THEN LOCATE PXR,PYR:PRINT" " ELSE PE
N 3:LOCATE PXR,PYR:PRINT"-" ELSE PEN 1:L
OCATE PXR,PYR:PRINT CHR$(127)'1ERE POS C
HOISIE
1280 PXR=PXR1:PYR=PYR1:GOTO 1300
1290 IF C(PXR1,PYR1)<>5 THEN IF C(PXR1,P
YR1)<>4 THEN LOCATE PXR1,PYR1:PRINT" "'E
LSE PEN 3:LOCATE PXR1,PYR1:PRINT"-" ELSE
 PEN 1:LOCATE PXR1,PYR1:PRINT CHR$(127)'
2EME POS CHOISIE
1300 VI=VI-1:GOSUB 8000
1310 REM TEST SUR DECISION DE RAVITAILLE
MENT
1320 IF C(PXR,PYR)<>4 THEN 1340'ABSENCE
D'ODEUR VICTUAILLES
1330 IF VI>PXR-PXM+3 THEN 1340 ELSE MVI=
M+1'RESERVE VITALE ROBOT SUFFISANTE,OU S
IGNALMT DECISION RAVITAILLEMENT LORS PRO
CHAINE BOUCLE
1340 IF C(PPXR,PPYR)<>3 THEN 1350 ELSE P
EN 2:LOCATE PPXR,PPYR:PRINT"^":GOTO 1370
1350 IF C(PPXR,PPYR)<>4 THEN 1360 ELSE P
EN 3:LOCATE PPXR,PPYR:PRINT"-":GOTO 1370
1360 PEN 3:LOCATE PPXR,PPYR:PRINT CHR$(3
3)':C(PPXR,PPYR)=RG+6'AFF ET REPRESENTAT
ION TRACE
1370 PEN 1:LOCATE PXR,PYR:PRINT CHR$(33)
'AFF ROBOT
1380 IF SONO=1 THEN SONO=0:FOR AT=1 TO 3
00:NEXT:GOTO 1390 ELSE SOUND 1,40,5,7
1390 GOSUB 4000'TEST MAITRE REJOINT
1400 GOSUB 6000'TEST TRAPPE
1410  PPXR=PXR:PPYR=PYR'FUTURE TRACE
1420 IF VI>0 THEN 1430 ELSE M=50'TEST PO
SSIBILITE CONTINUER SI ROBOT NON EPUISE
1430 FOR AT=1 TO 100:NEXT:NEXT M
1440 REM PARCOURS RATE
1450 SOUND 1,500,40,7:RG=RG+1:FOR AT=1 T
O 500:NEXT::PEN 3:LOCATE PXR,PYR:PRINT C
HR$(33)'AFF TRACE
1460 IF RG=8 THEN 110 ELSE VI=VI+25:GOSU
B 8000
```

```
1470 C(PXS1,PYS1)=0:C(PXS2,PYS2)=0:LOCAT
E PXS1,PYS1:PRINT" ":LOCATE PXS2,PYS2:PR
INT" ":GOTO 290
1480 END
2000 REM ZONE DES SOUS-PROGRAMMES
3000 REM SSPROG HEURT PILIER ET/OU RETOU
R EN ARRIERE
3010 OBS=0:RET=0
3020 IF C(PXR,PYR)=1 THEN 3030 ELSE 3040
'TEST PILIER
3030 SOUND 1,50,5,7:PEN 2:LOCATE PXR,PYR
:PRINT"*":FOR AT=1 TO 30:NEXT:PEN 1:LOCA
TE PXR,PYR:PRINT"=":OBS=1:GOTO 3060
3040 IF C(PXR,PYR)=RG+6 THEN 3050 ELSE R
ETURN'TEST TRACE
3050 IF TT>7 THEN RETURN ELSE SOUND 1,20
,5,7:PEN 2:LOCATE PXR,PYR:PRINT"*":FOR A
T=1 TO 30:NEXT:PEN 3:LOCATE PXR,PYR:PRIN
T CHR$(33):RET=1
3060 FOR AT=1 TO 50:NEXT:TT=TT+1:MPI=M'M
EMO NO BOUCLE
3070 REM ESSAI NOUVELLE DIRECTION
3080 REM HEURISTIQUE DE DEGAGEMENT (F):E
XPLORATION ORDONNEE DES  DIRECTIONS EN P
ARTANT DU PILIER HEURTE
3090 FOR AT=1 TO 50:NEXT AT:IF TT>1 THEN
 3230'CAS 2EME PASSAGE ET SUIVANTS
3100 IF VXR=-1 THEN 3130'CAS 1ER PASSAGE
,DETERMINATION DE LA DIRECTION D1 DU 1ER
 HEURT
3110 IF VXR=0 THEN 3160
3120 IF VXR =1 THEN 3180
3130 IF VYR=-1 THEN D1=5:GOTO 3210
3140 IF VYR=0 THEN D1=6:GOTO 3210
3150 IF VYR=1 THEN D1=7:GOTO 3210
3160 IF VYR=-1 THEN D1=4:GOTO 3210
3170 IF VYR=1 THEN D1=8:GOTO 3210
3180 IF VYR=-1 THEN D1=3:GOTO 3210
3190 IF VYR=0 THEN D1=2:GOTO 3210
3200 IF VYR=1 THEN D1=1
3210 D=D1'DIRECTION D DEBUT DE L'EXPLORA
TION
3220 SS=INT(2*RND)'TIRAGE SENS DE L'EXPL
ORATION
3230 REM EXPLORATION POSITION SUIVANTE
3240 IF SS=0 THEN D=D+1:IF D<9 THEN 3260
 ELSE D=D-8:GOTO 3260
3250 D=D-1:IF D>0 THEN 3260 ELSE D=D+8
3260 IF D=1 THEN 3270 ELSE IF D=2 THEN 3
280 ELSE IF D=3 THEN 3290 ELSE IF D=4 TH
EN 3300 ELSE IF D=5 THEN 3310 ELSE IF D=
6 THEN 3320 ELSE IF D=7 THEN 3330 ELSE 3
340
3270 VXR=1:VYR=1:GOTO 3350
```

```
3280 VXR=1:VYR=0:GOTO 3350
3290 VXR=1:VYR=-1:GOTO 3350
3300 VXR=0:VYR=-1:GOTO 3350
3310 VXR=-1:VYR=-1:GOTO 3350
3320 VXR=-1:VYR=0:GOTO 3350
3330 VXR=-1:VYR=1:GOTO 3350
3340 VXR=0:VYR=1
3350 RETURN
4000 REM SSPROG MAITRE REJOINT
4010 IF C(PXR,PYR)<>2 THEN RETURN
4020 FOR FOIS=1 TO 4:SOUND 1,27,5,7:FOR
AT=1 TO 100:NEXT:NEXT
4030 FOR AT=1 TO 800:NEXT:RG=RG+1:IF RG=
8 THEN 110 ELSE PEN 2:LOCATE PXM,PYM:PRI
NT CHR$(250):VI=VI+15:GOSUB 8000
4040 C(PXS1,PYS1)=0:LOCATE PXS1,PYS1:PRI
NT" ":C(PXS2,PYS2)=0:LOCATE PXS2,PYS2:PR
INT" ":GOTO 290
5000 REM SSPROG RAVITAILLEMENT EN  VICTU
AILLES
5010 REM 1ERE PHASE:RECHERCHE EMPLACEMEN
T EXACT VICTUAILLES
5020 SOUND 1,100,30,5:SOUND 1,60,10,7::F
OR A=PPXR-1 TO PPXR+1:FOR B=PPYR-1 TO PP
YR+1:IF C(A,B)<>3 THEN 5040 ELSE PXR=A:P
YR=B:MVI=50:VI=VI+20:GOSUB 8000
5030 A=PPXR+1:B=PPYR+1:'SORTIE BOUCLE DE
 RECHERCHE
5040 NEXT B
5050 NEXT A
5060 REM 2EME PHASE:VICTUAILLES DEVOREES
5070 REM ELIM VICTUAILLES ET ODEUR DE L'
ENTOURAGE
5080 FOR A1=PXR-1 TO PXR+1:FOR B1=PYR-1
TO PYR+1
5090 C(A1,B1)=0
5100 IF A1=PPXR THEN 5130
5110 IF A1=PXR THEN 5150
5120 GOTO 5160
5130 IF B1=PPYR THEN 5170
5140 GOTO 5110
5150 IF B1=PYR THEN 5170
5160 LOCATE A1,B1:PRINT" "
5170 NEXT B1
5180 NEXT A1
5190 SONO=1:SOUND 1,1000,20,7:SOUND 1,80
0,30,6:SOUND 1,2000,30,7:FOR AT=1 TO 500
:NEXT:RETURN
6000 REM SSPROG TRAPPE
6010 IF C(PXR,PYR)<>5 THEN 6070
6020 FOR FOIS=1 TO 4
6030 SOUND 1,1500,10,7:SOUND 1,2500,10,6
:VI=VI-1:GOSUB 8000
6040 PEN 3:LOCATE PXR,PYR:PRINT CHR$(33)
```

```
:FOR AT=1 TO 200:NEXT:PEN 2:LOCATE PXR,P
YR:PRINT CHR$(127)
6050 NEXT FOIS
6060 PXR=PXR-1:PEN 1:LOCATE PXR,PYR:PRIN
T CHR$(33):FOR AT=1 TO 700:NEXT
6070 RETURN
7000 REM SSPROG SONNERIE
7010 IF VI>7 THEN RETURN ELSE FOR FOIS=1
 TO 30:SOUND 1,35,2,7:FOR AT=1 TO 30:NEX
T:NEXT FOIS:FOR AT=1 TO 500:NEXT AT
7020 NS=INT(2*RND)+1:IF NS=1 THEN 7040
7030 PEN 2:LOCATE PXS2,PYS2:PRINT"$":X=P
XS2:Y=24:GOTO 7050'CAS SONNERIE NO2 CONC
ERNEE
7040 PEN 2:LOCATE PXS1,PYS1:PRINT"$":X=P
XS1:Y=3'CAS SONNERIE NO1 CONCERNEE
7050 ES=17:TENT=0'1ERE TENTATIVE DE LOCA
LISATION DE LA SONNERIE
7060 PEN 2:LOCATE ES,Y:PRINT"?"'AFF POSI
TION SUPPOSE
7070 FOR AT=1 TO 250:NEXT AT:IF ES<>X TH
EN 7120'MAUVAISE LOCALISATION,RENVOI A U
NE NOUVELLE TENTATIVE
7080 IF C(ES,Y)=6 THEN SOUND 1,35,40,7::
VI=VI+10:GOSUB 8000:GOTO 7100'GAGNE
7090 SOUND 1,1500,50,7'PERDU
7100 LOCATE 1,3:PRINT"
                    ":LOCATE 1,4:PRINT"
                                    ":LOCATE 1
,23:PRINT"
     ":LOCATE 1,24:PRINT"
                        "
7110 PEN 1:LOCATE PXS1,PYS1:PRINT"S":LOC
ATE PXS2,PYS2:PRINT"S":FOR AT=1 TO 100:N
EXT:RETURN
7120 TENT=TENT+1'TENTATIVE SUIVANTE DE L
OCALISATION
7130 IF ES>X THEN ES=ES-16/(2^TENT) ELSE
 ES =ES+16/(2^TENT)
7140 GOTO 7060'RENVOI VERDICT
8000 REM SSPROG AFF VI
8010 PEN 1:LOCATE 31,2:PRINT"VI=";VI;" "
:RETURN
```

Commentaires techniques

- Les REM échelonnées dans le programme donnent les explications essentielles.
- MVI : numéro de la boucle de consommation de victuailles.

- VI : potentiel vital du robot, affiché à l'aide d'un sous-programme spécial.
- Le robot décide de se ravitailler si deux conditions sont simultanément remplies :

  Il est dans la zone odorante ;

  Il a faim : son potentiel vital est inférieur à la distance restant à parcourir pour rejoindre son maître.
- Sous-programme ravitaillement :

  Le robot est dans une case contiguë à celle des victuailles, il commence donc à explorer tactilement les huit cases alentour à l'aide d'une boucle FOR NEXT ;

  Il se met alors à table, et les victuailles, ainsi que la zone odorante, sont éliminées par remise à zéro de la fonction logique C(I, J) pour les neuf cases intéressées.
- Sous-programme sonneries :

  Le robot essaie de deviner l'emplacement exact de la sonnerie qui vient de retentir ; en effet, son ouïe lui permet seulement de savoir si elle se trouve près du bord supérieur ou inférieur de l'écran ; reste à en préciser la position horizontale PXS1 ou PXS2 suivant la sonnerie concernée ;

  Il procède par approches successives, en prenant chaque fois le milieu de l'intervalle que le programme lui désigne ;

  L'emplacement est ainsi rapidement trouvé, mais le robot ne gagne des points que si la sonnerie se trouve bien sur celle des deux rangées qu'il a explorée.

IX

L'AVENIR AU SERVICE DU PRÉSENT — L'ANTICIPATION

"L'avenir, c'est ce qui dépasse la main tendue."
L. ARAGON.

L'esprit, capable de plonger dans le passé pour en tirer des enseignements, est, chose beaucoup plus surprenante ne surprenant pourtant personne, également capable de se projeter dans l'avenir aux mêmes fins : tirer des enseignements d'événements non encore survenus...

La concrétisation, dans un programme d'ordinateur, de cette opération mentale rocambolesque si courante, *l'anticipation*, conduit ceux qui s'y risquent à une surprise d'un autre genre, dont la réalité incontournable résiste à toute contestation philosophique : il est nécessaire, pour anticiper, de *mémoriser l'avenir* en le traitant comme un présent fictif, qui devient donc aussitôt un passé fictif.

Imprégné de tout cela, une anticipation expérimentale consciemment vécue procure à celui qui s'y livre la délicieuse sensation d'un détriplement de la personnalité. Le lecteur masochiste pourra essayer par exemple de s'observer attentivement en même temps qu'il réfléchit à la meilleure manière de contourner un pilier au moyen de l'heuristique I de l'étape E28 ci-après.

Cet exercice est pourtant pratiqué en permanence dans la vie de tous les jours, mais ses mécanismes sont heureusement inconscients. La pratique de l'I.A. passe nécessairement par l'autopsie impitoyable de ce phénomène.

Les deux cas présentés dans ce chapitre sont relativement simples : d'une part, l'anticipation n'y est faite qu'*à la profondeur 1*, ce qui n'est déjà pas si mal, et d'autre part elle n'est pas soumise aux aléas du comportement d'un adversaire comme par exemple au jeu d'échecs. En effet, la cible de E27 est mobile mais son mouvement rectiligne uniforme est connu ; et la position du maître de E28 est fixe.

La recherche d'un comportement d'anticipation est assez simple, comme nous allons le voir. Par contre, la concrétisation de ce comportement dans un programme conduit rapidement à une grande complexité en raison de l'augmentation exponentielle du nombre de cas à considérer.

Aussi est-il assez plaisant, tous efforts terminés, de contempler le robot superman se livrer à des exploits dont les ressorts profonds finissent par échapper, par exemple en E29 où tout jalonnement est supprimé.

L'on comprend mieux l'extraordinaire performance que peut constituer la conduite intelligente d'une action quelconque lorsque, au-delà de l'imagination et de la logique qu'elle requiert, l'anticipation y introduit les prodigieuses possibilités de la concentration et de la mémoire.

ÉTAPE E27

UNE ANTICIPATION DE NAVIGATION... ET UN CONCOURS PROPOSÉ AU LECTEUR

Présentation

Nous retrouvons, dans cet exemple, à l'étape E5, un missile autoguidé poursuivant une cible mobile sur terrain nu.

Il n'y a pas d'obstacles, il n'y a donc pas d'heuristique de dégagement. Seule l'heuristique de navigation est en cause. En E5 c'était l'heuristique 2.

Cette fois, le missile améliore son comportement. Au lieu de se diriger vers la cible en tenant simplement compte du lieu où elle se trouve, il anticipe sur les futures positions de cette cible et se dirige *un peu en avant d'elle*.

Il a pour cela toutes les informations voulues, puisque la cible est animée d'un mouvement uniforme et que sa vitesse est connue. Elle reste constante entre deux rebondissements successifs.

Ces données sont censées être portées en permanence à la connaissance du missile qui peut donc les utiliser à son profit. Il est également doté d'un sens de la vue lui permettant d'apprécier la direction et la distance de la cible (en E5, il ne pouvait en apprécier que la direction).

L'heuristique utilisée est baptisée 6, elle constitue une amélioration de l'heuristique "ordinaire" 2.

Les deux relations simples qui fournissaient les deux composantes VXR et VYR de E5 conservent la même forme, mais comprennent un terme complémentaire d'aspect insolite dans lequel il est toutefois assez facile de discerner l'intervention de la vitesse de la cible (en grandeur et en direction) et de la distance à laquelle elle se trouve par rapport au missile. Cette distance constitue en quelque sorte le bras de levier qui doit amplifier l'effet de l'anticipation.

Curieusement, il s'agit donc de deux relations assez compliquées, mais dont le formalisme recouvre en fait une idée simple : précéder le mouvement de la cible. Ces relations sont bien de nature heuristique et non pas algorithmique, car elles ne découlent pas d'une analyse mathématique rigoureuse, et nul ne peut assurer qu'elles conduiront au succès ou même qu'elles seront supérieures à tous coups à l'heuristique ordinaire 2.

C'est une sorte de fonction d'évaluation de la situation de même nature intuitive que celle de E13, avec cependant une différence essentielle :

elle ne sert pas à comparer plusieurs positions en vue de choisir la meilleure, elle agit directement, de façon "intégrée", sur la vitesse du missile en fonction de l'ensemble des données de la situation (positions, vitesses). C'est une fonction d'évaluation de situation, et non plus seulement de position.

Cela est logique, s'agissant d'intervenir sur la navigation et non pas sur un dégagement , sur la stratégie et non pas sur la tactique, de rechercher le *mieux* et non pas le *moins mal*.

Le seul moyen d'apprécier la valeur de ces formules consiste à en vérifier l'efficacité sur le terrain. Pour ce faire, le programme ci-après met en compétition l'heuristique 2 et la nouvelle heuristique 6. Les deux missiles font tour à tour un pas, et que le meilleur gagne !

On peut choisir au départ la vitesse de la cible dans la fourchette 0,5 à 2 (cases par pas). Rappelons que la vitesse du missile est de une case par pas.

On constate rapidement que, pour des vitesses de cible inférieures ou égales à 1, les deux heuristiques se valent ; la puissance physique des missiles, égale ou supérieure à celle de la cible, les dispense de voir plus loin que le bout du nez de cette dernière.

Par contre, dès que la cible va un peu plus vite que les missiles (vitesses supérieures à 1 — par exemple 1,3), l'anticipation révèle toute sa valeur, l'intelligence domine la puissance matérielle.

Alors qu'en E5 la cible échappait souvent aux poursuites au-delà de 1,2 ou 1,3, elle est ici rattrapée à tous coups par le missile à anticipation, même pour des vitesses de cible atteignant 2.

Le score du match est affiché à l'écran pour permettre de suivre la gradation ; c'est dans les circonstances difficiles que les qualités s'affirment.

Le lecteur pourra essayer d'améliorer l'heuristique 6 par tous moyens que son imagination lui suggérera. Puis il pourra mesurer le résultat de ses efforts en laissant l'heuristique 6 au missile qui la possède et en dotant le missile ordinaire de la nouvelle heuristique de son cru.

Deux conditions sont simplement requises pour garantir l'honnêteté de toute nouvelle heuristique : seules les deux formules donnant VXMIO et VYMIO peuvent être manipulées, et ces deux composantes ne peuvent pas prendre de valeurs supérieures à 1 en valeur absolue (donc, pas plus d'une case par pas).

Il n'est pas nécessaire d'être mathématicien, mais il n'est pas mauvais de posséder un certain sens des formules.

Programme

```
100 'ETAPE E27:COMME E5 AVEC ANTICIPATIO
N,PAR LE MISSILE,DU COMPORTEMENT DE LA C
IBLE;MISSILE AUTOGUIDE,CIBLE MOBILE,TERR
AIN NU,HEUR NAV (6);COMPETITION AVEC LE
MISSILE "ORDINAIRE" DE E5 (HEUR (2)) EN
COULEUR PASTEL
110 INK 0,22:INK 1,1:INK 2,6:INK 3,23:CL
S:CLEAR:SCO=0:SCA=0'INIT SCORE
120 SYMBOL AFTER 32:SYMBOL 33,&F0,&E0,&F
0,&B8,&1C,&F,&6,&4
130 CLS:R=0:PEN 1:LOCATE 4,11:PRINT"VITE
SSE DE LA CIBLE (DE 0,5 A 2) ?":LOCATE 2
,13:PRINT"PUIS APPUYEZ SUR LA BARRE D'ES
PACEMENT"
140 D$=INKEY$:RANDOMIZE TIME:IF D$=" " T
HEN 150 ELSE IF D$<"." OR D$>"9" THEN SO
UND 1,30,5,7:FOR AT=1 TO 200:NEXT:GOTO 1
40 ELSE R=R+1:A$(R)=D$:PEN 1:LOCATE 17+R
,15:PRINT A$(R):SOUND 1,20,10,7:GOTO 140
150 K$=" ":FOR Z=1 TO R:K$=K$+A$(Z):NEXT
 Z:K=VAL(K$):ON ERROR GOTO 130
160 PEN 1:LOCATE 16,16:PRINT"VIT=";K
170 IF K<0.5 OR K>2 THEN SOUND 1,500,30,
7:FOR AT=1 TO 1000:NEXT:GOTO 130
180 FOR AT=1 TO 1000:NEXT:CLS:H=INT(2*RN
D):IF H=1 THEN PXC=INT(10*RND+8):PYC=6 E
LSE PXC=8:PYC=INT(10*RND+8)'TIRAGE POSIT
ION CIBLE
190 PPXC=PXC:PPYC=PYC
200 PXMI=INT(25*RND)+9:PYMI=INT10*RND+13
:PPXMI=PXMI:PPYMI=PYMI'POSITION DEPART M
ISSILE ET MISE EN RESERVE POUR EFFACEMEN
T
210 PXMIO=PXMI:PYMIO=PYMI:PPXMIO=PXMIO:P
PYMIO=PYMIO'POSITION DEPART MISSILE ORDI
NAIRE IDENTIQUE A CELLE DU PREMIER
220 PEN 2:LOCATE PXC,PYC:PRINT"o":PEN 1:
LOCATE PXMI,PYMI:PRINT CHR$(33):LOCATE 3
,1:PRINT"ORD=";SCO;"
   ";"ANT=";SCA'AFF CIBLE,MISSILES ET SC
ORE
230 VXC=K*SGN(2*RND-1):VYC=K*SGN(2*RND-1
)'TIRAGE COMPOSANTES VITESE CIBLE
240 SOUND 1,200,80,7:FOR AT=1 TO 2000:NE
XT'TOP DEPART
1000 REM BOUCLE DES MOUVEMENTS
1010 FOR M= 1 TO 500
1020 REM COMMANDE CHANGEMENT DE VITESE C
IBLE
1030 CH$=INKEY$:IF CH$<>" " THEN 1040 EL
SE 110
1040 REM MOUVMT CIBLE
```

```
1050 PXC=PXC+VXC:PYC=PYC+VYC'CALC NOUVEL
LE POS CIBLE
1060 GOSUB 3000' EFFETS DE BORDS
1070 PEN 2:IF CINT(PXC)<>40 OR CINT(PYC)
<>25 THEN LOCATE PPXC,PPYC:PRINT " ":LOC
ATE PXC,PYC:PRINT"o":SOUND 1,100,5,7:PPX
C=PXC:PPYC=PYC ELSE SOUND 1,53,10,7:PXC=
39:VXC=-ABS(VXC):VYC=-ABS(VYC):GOTO 1070
'AFF CIBLE (SAUF POS 40,25 INTERDITE) ET
 MISE EN RES
1080 GOSUB 5000'CIBLE ATTEINTE
1090 REM MOUVMT MISSILE ORDINAIRE,HEUR N
AV (2)
1100 VXMIO=SGN(CINT(PXC)-PXMIO):VYMIO=SG
N(CINT(PYC)-PYMIO):
1110 PXMIO=PXMIO+VXMIO:PYMIO=PYMIO+VYMIO
1120 IF PPXMIO<>PPXMI THEN 1130 ELSE IF
PPYMIO<>PPYMI THEN 1130  ELSE PEN 1:LOCA
TE PPXMIO,PPYMIO:PRINT CHR$(33):GOTO 114
0
1130 LOCATE PPXMIO,PPYMIO:PRINT" "
1140 PEN 3:LOCATE PXMIO,PYMIO:PRINT CHR$
(33):SOUND 1,30,5,7:PPXMIO=PXMIO:PPYMIO=
PYMIO
1150 REM MOUVMT MISSILE A ANTICIPATION,H
EURIST NAV (2) AMELIOREE
1160 VXMI=SGN(CINT(PXC)-PXMI+VXC*ABS(ABS
(CINT(PYC)-PYMI)-1)):VYMI=SGN(CINT(PYC)-
PYMI+VYC*ABS(ABS(CINT(PXC)-PXMI)-1))'ADJ
ONCTION D'UN TERME COMPLEMENTAIRE A L'HE
URIST (2) POUR TENIR COMPTE DE LA VITESS
E DE LA CIBLE
1170 PXMI=PPXMI+VXMI:PYMI=PPYMI+VYMI
1180 GOSUB 4000'EFFETS DE BORDS
1190 LOCATE PPXMI,PPYMI:PRINT" ":PEN 1:L
OCATE PXMI,PYMI:PRINT CHR$(33):SOUND 1,3
0,1,7:PPXMI=PXMI:PPYMI=PYMI'AFF ET MISE
EN RES
1200 GOSUB 5000'TEST CIBLE ATTEINTE

1210 FOR AT=1 TO 50:NEXT:NEXT M

1220 END
2000 REM ZONE DES SOUS-PROGRAMMES
3000 REM SSPROG EFFETS DE BORDS CIBLE
3010 IF CINT(PXC) >0 THEN 3020 ELSE SOUN
D 1,53,10,7:PXC=1:VXC=-VXC
3020 IF CINT(PXC)< 41 THEN 3030 ELSE SOU
ND 1,53,10,7:PXC=40:VXC=-VXC
3030 IF CINT(PYC)>0 THEN 3040 ELSE SOUND
 1,53,10,7:PYC=1:VYC=-VYC:RETURN
3040 IF CINT(PYC)<26 THEN RETURN ELSE SO
UND 1,53,10,7:PYC=25:VYC=-VYC:RETURN
4000 REM SSPROG EFFETS DE BORDS MISSILE
```

```
A ANTICIPATION
4010 IF PXMI>0 THEN 4020 ELSE PXMI=1:GOT
O 4030
4020 IF PXMI<41  THEN 4030 ELSE PXMI=40
4030 IF PYMI>0 THEN 4040 ELSE PYMI=1:RET
URN
4040 IF PYMI<26 THEN RETURN ELSE PYMI=25
:RETURN
5000 REM SSPROG CIBLE ATTEINTE
5010 IF PXMIO<>CINT(PXC) THEN 5020 ELSE
IF PYMIO<>CINT(PYC)  THEN 5020 ELSE PEN
2:LOCATE PXMIO,PYMIO:PRINT CHR$(33):SCO=
SCO+1:GOTO 5030
5020 IF PXMI<>CINT(PXC)THEN RETURN ELSE
IF PYMI<>CINT(PYC) THEN RETURN ELSE PEN
2:LOCATE PXMI,PYMI:PRINT CHR$(33):SCA=SC
A+1
5030 FOR FOIS=1 TO 4:SOUND 1,27,5,7:FOR
AT=1 TO 100:NEXT:NEXT:FOR AT=1 TO 800:NE
XT:GOTO 180
```

Commentaires techniques

- Il y a deux missiles dans la boucle de mouvement : le missile ordinaire, heuristique 2, et le missile à anticipation, heuristique 6.
- Le premier est affiché en teinte atténuée, le second en bleu.
- On distingue facilement le terme complémentaire de l'heuristique 6, en comparant les deux relations donnant les composantes VXMIO et VYMIO d'une part, et VXMI et VYMI d'autre part.
- La vitesse de la cible est introduite en début de programme à l'aide d'une instruction INPUT protégée par des garde-fous contre les erreurs de clavier. Après avoir formé la valeur de la vitesse, qui s'affiche progressivement sur l'écran, on appuie sur la barre d'espacement pour lancer la poursuite.
- L'heuristique 2 ne nécessite pas d'effets de bords pour le missile ordinaire qui reste toujours dans les limites d'écran.
- Par contre, les effets de bords sont indispensables pour l'heuristique 6 du missile à anticipation.
- Lorsque les deux missiles atteignent la cible en même temps, la marque est magnanimement attribuée au missile ordinaire.

ÉTAPE E28

UNE ANTICIPATION DE DÉGAGEMENT À PROFONDEUR 1

Présentation

Cette fois, c'est l'étape E13 qui servira de base, avec son robot, son maître, ses piliers, et ses heuristiques 1 et F. Et c'est l'opération de dégagement F qui bénéficiera d'une anticipation, la nouvelle heuristique obtenue étant alors baptisée I.

Le principe en est simple. Lorsqu'il arrive sur un pilier, on sait que le robot, en appliquant l'heuristique F, trouve deux positions libres, l'une à droite et l'autre à gauche de l'obstacle.

Mais au lieu de choisir comme en E13 la meilleure (ce qui constituait déjà une forme d'anticipation puisqu'il ne se déplaçait qu'après avoir repéré à blanc les deux positions possibles), il pousse plus loin ses simulations, son anticipation : de chacune des deux positions ainsi obtenues, il simule un pas supplémentaire au cours duquel il peut avoir ou non à contourner un autre pilier.

On dit qu'il anticipe *à la profondeur 1*, alors qu'en E13 il ne s'agissait que de la profondeur 0. On dit aussi qu'il avance de deux branches sur *l'arbre des mouvements* [1].

Il aboutit ainsi, dans le cas général, à *quatre positions possibles*, deux pour chacune des deux positions précédentes. Mais il ne rencontre pas forcément de pilier dans cette seconde tournée de simulations. Il peut ainsi aboutir, non pas à quatre, mais à trois ou à deux positions de profondeur 1. C'est parmi elles que, à l'aide de la fonction d'évaluation déjà étudiée en E13, il choisit la meilleure et d'où il déduit la position de profondeur 0 à occuper.

Il procède ainsi avant chacun de ses pas, tout au long du trajet. En résumé :

- en l'absence de pilier, le robot avance en utilisant son heuristique de navigation ;
- au contact d'un pilier, il fait jouer son heuristique de dégagement F et trouve deux positions libres A et B possibles ;

1. En fait, cet arbre est déjà bien élagué, puisque l'heuristique F n'en a conservé que deux branches.

- puis, à partir de chacune, il simule un pas en utilisant ses mêmes heuristiques 1 ou F suivant qu'il y a ou non un obstacle ;
- il trouve ainsi, en général, quatre positions A1, A2, B1, B2 ;
- il les évalue et choisit la meilleure ;
- enfin il occupe la position A ou B qui a donné accès à cette meilleure.

L'efficacité du procédé est très grande.

C'est un principe de même nature qui préside à l'élaboration du programme de jeux d'échecs, avec une complication supplémentaire : il faut anticiper, non seulement sur son prochain coup, mais aussi sur celui de l'adversaire (méthode dite du minimax). Les meilleurs programmes se fondent, d'une part sur la plus grande profondeur d'anticipation, et d'autre part sur la meilleure qualité de la fonction d'évaluation.

Le programme démonstratif ci-après visualise par des points rouges les deux positions A et B de profondeur 0, et par des points d'interrogation les quatre (ou trois, ou deux) positions A1, A2, B1 et B2 de profondeur 1.

Il arrive bien entendu que A1 et A2, ou B1 et B2, soient confondus s'il n'y a pas de pilier à contourner au cours de la seconde tournée de simulations.

Le programme met par ailleurs en évidence la gymnastique assez complexe des instructions à mettre en œuvre dans l'ordinateur pour simuler toutes ces simulations...

On voit aussi que, pour décrire des situations et des opérations de ce genre, la langue naturelle, aussi déliée soit-elle, trouve rapidement ses limites d'expression.

Seul un langage plus rigoureux, dont la rigueur est d'ailleurs obtenue par un formalisme et par des règles de manipulation intransigeantes et rigides, peut surmonter les difficultés de représentation et de traitement.

Le lecteur pourra essayer d'aller plus loin et d'atteindre la profondeur 2, obtenant donc huit positions possibles sur l'arbre des mouvements. L'exercice ressemble alors à celui du jongleur qui doit simultanément maîtriser les trajectoires de ses anneaux, de ses balles et de ses torches, mais quelle splendeur dans le spectacle, pour qui devine les ressorts profonds !

Rien n'interdit en théorie d'anticiper le meilleur parcours *complet* possible : il suffit d'aller à la profondeur 30, c'est-à-dire de simuler 2 puissance 31 possibilités, soit un peu plus de deux milliards... Il est moins fatigant, plus raisonnable , plus heuristique, de renoncer à cette débauche combinatoire et de laisser Roméo se livrer à ses promenades de E23, quitte à lui en faire accomplir une centaine : il est *à peu près sûr* (cette formule pourrait devenir la devise des heuristiciens) que l'une d'elles, judicieusement retenue par le robot, serait la perle convoitée.

Programme

```
100 'ETAPE E28:COMME E13 AVEC ANTICIPATI
ON "A PROFONDEUR 1" DE L'HEURISTIQUE DE
DEGAGMT (F) POUR OBTENIR L'HEURISTIQUE (
I)
110 INK 0,22:INK 1,1:INK 2,6:INK 3,14:CL
S:CLEAR
120 DIM C(33,25)'TABLEAU DES COORDONNEES
 DES CASES DE L'ECRAN SUSCEPTIBLES DE RE
CEVOIR UN PILIER
130 SYMBOL AFTER 32:SYMBOL 33,&0,&7C,&D6
,&6C,&38,&FE,&7C,&6C'DESSIN ROBOT;DESSIN
 MAITRE:CODE 250
140 REM NB ALEATOIRES VARIES
150 RANDOMIZE TIME
160 RG=1:SIM=0'INIT RANG DU PARCOURS RG
ET RANG DE LA SIMULATION SIM
170 CLS:H=INT(2*RND):IF H=1 THEN PXM=INT
(25*RND+6):PYM=6 ELSE PXM=6:PYM=INT(12*R
ND+6)'TIRAGE POSITION MAITRE
180 PXR=36:PYR=21:PPXR=PXR:PPYR=PYR:PPXR
T=PXR:PPYRT=PYR'POS DEPART ROBOT,MISE EN
 RESERVE POUR CALCULS DE SIMULATION (PPX
R,PPYR) ET MISE EN RES POUR TRACE
190 PEN 1:LOCATE PXR,PYR:PRINT CHR$(33):
PEN 2:LOCATE PXM,PYM:PRINT CHR$(250)'AFF
 ROBOT ET MAITRE
200 IF RG>1 THEN 280
210 FOR N=0 TO 30:TERR$=INKEY$:IF TERR$<
>" " THEN 220 ELSE 110
220 I=INT((33-PXM)*RND)+PXM-1:J=INT((21-
PYM)/(36-PXM)*(I-PXM))+INT((5*RND-2)*(I-
PXM)/8)+PYM+1
230 IF J>PYM-2 THEN 250
240 J=PYM-1:GOTO 270
250 IF J<>PYM THEN IF J<22 THEN 270 ELSE
 J=21:GOTO 270
260 IF I<>PXM THEN 270 ELSE J=PYM-1
270 PEN 1:LOCATE I,J:PRINT"=":C(I,J)=1:N
EXT'AFF ET REPRESENTATION PILIER
280 SOUND 1,200,80,7:FOR AT=1 TO 2000:NE
XT'TOP DEPART
1000 REM BOUCLE DES MOUVEMENTS
1010 FOR M=1 TO 50'50 MOUVMTS MAXI ADMIS
 POUR ATTEINDRE MAITRE
1020 TERR$=INKEY$:IF TERR$<>" " THEN 103
0 ELSE 110
1030 IF PXR<>PXM THEN NUME=ABS(PYR-PYM):
DENO=ABS(PXR-PXM):IN=NUME/DENO ELSE 1060
'HEUR NAV (1)
1040 IF IN<0.5 THEN VXR=SGN(PXM-PXR):VYR
=0:GOTO 1070
1050 IF IN>2 THEN VXR=0:VYR=SGN(PYM-PYR)
```

```
:GOTO 1070
1060 VXR=SGN(PXM-PXR):VYR=SGN(PYM-PYR)
1070 TT=0'INITIALISATION DU RANG TT DES
TENTATIVES POUR TROUVER UNE POSITION LIB
RE APRES HEURT PILIER OU RETOUR EN ARRIE
RE
1080 PXR=PPXR+VXR:PYR=PPYR+VYR
1090 IF PXR>33 THEN 1610 ELSE GOSUB 3000
'TEST HEURT PILIER
1100 IF OBS=0 THEN 1110 ELSE 1080'RECALC
UL POS
1110 IF RET=0 THEN 1120 ELSE 1080
1120 IF M<>MPI THEN 1360
1130 REM HEURIST DE DEGT (F):ESSAI AUTRE
 SENS D'EXPLORATION
1140 IF ESS=1 THEN 1190 ELSE PXR1=PXR:PY
R1=PYR:VXR1=VXR:VYR1=VYR'MEMO 1ERE POS (
A SI SIM=0,A1 SI SIM=1,B1 SI SIM=2)
1150 IF SIM=0 THEN PEN 3:LOCATE PXR,PYR:
PRINT"-" ELSE PEN 2:LOCATE PXR,PYR:PRINT
"?"
1160 FOR AT=1 TO 100:NEXT:SSS=SS:IF SSS=
1 THEN SS=0 ELSE SS=1'INVERSION SENS DE
ROTATION
1170 TT=2:ESS=1:D=D1:GOSUB 3250
1180 GOTO 1080'RECALCUL POS
1190 ESS=0:MPI=0'REINIT ESS ET MPI
1200 PXR2=PXR:PYR2=PYR:VXR2=VXR:VYR2=VYR
'MEMO 2EME POSITION (B SI SIM=0,A2 SI SI
M=1,B2 SI SIM=2)
1210 IF SIM=0 THEN PEN 3:LOCATE PXR,PYR:
PRINT"-" ELSE PEN 2:LOCATE PXR,PYR:PRINT
"?"
1220 REM CALCUL DES 2 DISTANCES DIST1 ET
 DIST2 AU MAITRE
1230 IF SIM=0 THEN 1280'VERS LE DEBUT DE
S SIMULATIONS
1240 HOR1=ABS(PXM-PXR1):VERT1=ABS(PYM-PY
R1)::IF HOR1>VERT1 THEN DIST1=HOR1 ELSE
DIST1=VERT1
1250 HOR2=ABS(PXM-PXR):VERT2=ABS(PYM-PYR
):IF HOR2>VERT2 THEN DIST2=HOR2 ELSE DIS
T2=VERT2
1260 IF SIM=2 THEN 1270 ELSE DISTA1=DIST
1:DISTA2=DIST2:PXRA1=PXR1:PYRA1=PYR1:PXR
A2=PXR2:PYRA2=PYR2:PI1=1:GOTO 1280'CAS S
IM=1:2 POS A1 ET A2 OBT A PARTIR DE A,ME
MO DISTCES ,MEMO POSITS POUR EFFT ULT ET
 SIGNALMT PIL RENC AU COURS 1ERE SIM
1270 DISTB1=DIST1:DISTB2=DIST2:PI2=1:GOT
O 1410'CAS SIM=2:2 POS  OBT A PARTIR DE
B,MEMO DISTCES ET SIGNALMT PIL RENC AU C
OURS 2EME SIM;INUTILE MEMORISER POS PXR1
/PYR1 ET PXR2/PYR2 QUI VONT ETRE UTILISE
```

```
ES TELLES QUELLES
1280 SIM=SIM+1:IF SIM=2 THEN 1340
1290 REM SIM=1:MEMO DES 2 POS A ET B D'O
U VONT SE FAIRE LES 2 SIM 1 ET 2 DU PAS
SUIVT DU ROBOT,ET DES VITESSES CORRESPON
DANTES POUR EVITER LES RET ARR
1300 PXRA=PXR1:PYRA=PYR1:VXRA=VXR1:VYRA=
VYR1:PXRB=PXR2:PYRB=PYR2:VXRB=VXR2:VYRB=
VYR2
1310 M=M-2'RETROGRAD M POUR COMPENSER LE
S 2 BOUCLES DE SIMULATION ET EVITER FIN
DE PARCOURS PREMATUREE (M>50)
1320 REM 1ERE SIM (A PARTIR DE A)
1330 PXR=PXRA:PYR=PYRA:VXR=VXRA:VYR=VYRA
:GOTO 1630
1340 REM 2EME SIM (A PARTIR DE B)
1350 PXR=PXRB:PYR=PYRB:VXR=VXRB:VYR=VYRB
:GOTO 1630
1360 REM PAS DE PILIER
1370 IF SIM=0 THEN 1610 ELSE PXR3=PXR:PY
R3=PYR:PEN 2:LOCATE PXR,PYR:PRINT"?":HOR
3=ABS(PXM-PXR3):VERT3=ABS(PYM-PYR3)'CAS
SIM=1 OU 2
1380 IF HOR3<VERT3 THEN DIST3=VERT3 ELSE
 DIST3=HOR3
1390 IF SIM=1 THEN DISTA1=DIST3:DISTA2=1
00:PXRA3=PXR3:PYRA3=PYR3:PI1=0:GOTO 1280
'POS A1 OBT A PARTIR DE A:MEMO DISTA1,DI
STA2 FICTIVE + ELEVEE,MEMO POS POUR EFFT
 ULT,SIGNALMT PAS DE PIL AU COURS 1ERE S
IM
1400 IF SIM=2 THEN DISTB1=DIST3:DISTB2=1
00:PI2=0'COMME 1390 MAIS INUTILE MEMO PO
S CAR PXR3/PYR3 SERONT UTILISES TELS QUE
LS
1410 REM CHOIX PLUS COURTE DISTANCE AU M
AITRE DES 4 POS A1,A2,B1 ET B2
1420 IF DISTA1<DISTA2 THEN 1470
1430 IF DISTA2<DISTB1 THEN 1460
1440 IF DISTB1<DISTB2 THEN 1530
1450 DIST=DISTB2 :GOTO 1550
1460 IF DISTA2<DISTB2 THEN 1510
1470 IF DISTA1<DISTB1 THEN 1490
1480 GOTO 1440
1490 IF DISTA1<DISTB2 THEN 1520
1500 GOTO 1450
1510 DIST=DISTA2:GOTO 1540
1520 DIST=DISTA1:GOTO 1540
1530 DIST=DISTB1:GOTO 1550
1540 PXR=PXRA:PYR=PYRA:VXR=VXRA:VYR=VYRA
:GOTO 1560
1550 PXR=PXRB:PYR=PYRB:VXR=VXRB*VYR=VYRB
1560 REM REINIT SIM ET EFFT POS PROVISOI
RES
```

```
1570 FOR AT=1 TO 300:NEXT:SIM=0:LOCATE P
XRA,PYRA:PRINT" ":LOCATE PXRB,PYRB:PRINT
" "
1580 IF PI1=0 THEN LOCATE PXRA3,PYRA3:PR
INT" " ELSE LOCATE PXRA1,PYRA1:PRINT" ":
LOCATE PXRA2,PYRA2:PRINT" "
1590 IF PI2=0 THEN LOCATE PXR3,PYR3:PRIN
T" " ELSE LOCATE PXR1,PYR1:PRINT" ":LOCA
TE PXR2,PYR2:PRINT" "
1600 IF DIST<>0 THEN 1610 ELSE PEN 2:LOC
ATE PXM,PYM:PRINT CHR$(250)
1610 PEN 1:LOCATE 31,1:PRINT M;"":PEN 3:
LOCATE PPXRT,PPYRT:PRINT CHR$(33):SOUND
1,35,5,7:PEN 1:LOCATE PXR,PYR:PRINT CHR$
(33)'AFF TRACE ET ROBOT
1620 PPXRT=PXR:PPYRT=PYR'MEMO POS POUR T
RACE ULT
1630 PPXR=PXR:PPYR=PYR:VVXR=VXR:VVYR=VYR
'MEMO POS ET VIT POUR CALCULS ULT
1640 IF SIM<>0 THEN 1650 ELSE GOSUB 4000
'TEST MAITRE REJOINT (SAUF PENDANT LES S
IMULATIONS)
1650 FOR N=1 TO 40:NEXT:NEXT M
1660 SOUND 1,500,40,7:RG=RG+1:IF RG=6 TH
EN 110 ELSE PEN 3:LOCATE PXR,PYR:PRINT"C
HR$(33):FOR AT=1 TO 1000:NEXT:GOTO 180'P
ARCOURS NON REUSSI
1670 END
2000 REM ZONE DES SOUS-PROGRAMMES
3000 REM SSPROG HEURT PILIER ET/OU RETOU
R EN ARRIERE
3010 OBS=0:RET=0
3020 REM TEST PILIER
3030 IF C(PXR,PYR)=1 THEN 3040 ELSE 3050
3040 FOR AT=1 TO 50:NEXT:SOUND 1,50,5,7:
PEN 2:LOCATE PXR,PYR:PRINT"*":FOR AT=1 T
O 50:NEXT:PEN 1:LOCATE PXR,PYR:PRINT"=":
OBS=1:GOTO 3090
3050 REM TEST RETOUR ARRIERE
3060 IF VXR=-VVXR THEN IF VYR=-VVYR THEN
 3070 ELSE RETURN ELSE RETURN
3070 IF TT>7 THEN RETURN ELSE FOR AT=1 T
O 50:NEXT AT:SOUND 1,20,5,7:PEN 2:LOCATE
 PXR,PYR:PRINT"*":FOR AT=1 TO 50:NEXT:PE
N 3:LOCATE PXR,PYR:PRINT CHR$(33):RET=1
3080 IF PXR<>PPXRT OR PYR<>PPYRT THEN PE
N 3:LOCATE PXR,PYR:PRINT CHR$(33) ELSE P
EN 1:LOCATE PXR,PYR:PRINT CHR$(33)
3090 FOR AT=1 TO 100:NEXT:TT=TT+1:MPI=M'
MEMO NO BOUCLE
3100 REM ESSAI NOUVELLE DIRECTION:HEURIS
T DEGMT (I)
3110 FOR AT=1 TO 100:NEXT AT:IF TT>1 THE
N 3250'CAS 2EME PASSAGE ET SUIVANTS
```

```
3120 IF VXR=-1 THEN 3150'CAS 1ER PASSAGE
,DETERMINATION DE LA DIRECTION D1 DU 1ER
 HEURT
3130 IF VXR=0 THEN 3180
3140 IF VXR =1 THEN 3200
3150 IF VYR=-1 THEN D1=5:GOTO 3230
3160 IF VYR=0 THEN D1=6:GOTO 3230
3170 IF VYR=1 THEN D1=7:GOTO 3230
3180 IF VYR=-1 THEN D1=4:GOTO 3230
3190 IF VYR=1 THEN D1=8:GOTO 3230
3200 IF VYR=-1 THEN D1=3:GOTO 3230
3210 IF VYR=0 THEN D1=2:GOTO 3230
3220 IF VYR=1 THEN D1=1
3230 D=D1'DIRECTION D DEBUT DE L'EXPLORA
TION
3240 SS=INT(2*RND)'TIRAGE SENS DE L'EXPL
ORATION
3250 REM EXPLORATION POSITION SUIVANTE
3260 IF SS=0 THEN D=D+1:IF D<9 THEN 3280
 ELSE D=D-8:GOTO 3280
3270 D=D-1:IF D>0 THEN 3280 ELSE D=D+8
3280 IF D=1 THEN 3290 ELSE IF D=2 THEN 3
300 ELSE IF D=3 THEN 3310 ELSE IF D=4 TH
EN 3320 ELSE IF D=5 THEN 3330 ELSE IF D=
6 THEN 3340 ELSE IF D=7 THEN 3350 ELSE 3
360
3290 VXR=1:VYR=1:GOTO 3370
3300 VXR=1:VYR=0:GOTO 3370
3310 VXR=1:VYR=-1:GOTO 3370
3320 VXR=0:VYR=-1:GOTO 3370
3330 VXR=-1:VYR=-1:GOTO 3370
3340 VXR=-1:VYR=0:GOTO 3370
3350 VXR=-1:VYR=1:GOTO 3370
3360 VXR=0:VYR=1
3370 RETURN
4000 REM SSPROG MAITRE REJOINT
4010 IF PXR<>PXM THEN RETURN ELSE IF PYR
<>PYM THEN RETURN
4020 FOR FOIS=1 TO 4:SOUND 1,27,5,7:FOR
AT=1 TO 100:NEXT:NEXT
4030 FOR AT=1 TO 800:NEXT:RG=RG+1:IF RG<
5 THEN PEN 3:LOCATE PPXR,PPYR:PRINT CHR$
(33):GOTO 180 ELSE 110
```

Commentaires techniques

- Des REM ont été placées en lieux opportuns pour éclairer le cheminement des opérations d'anticipation.
- Tant que Roméo ne rencontre pas de pilier, le programme se déroule comme en E13 : OBS et RET = 0, avancement d'un pas suivant l'heuristique 1.

- Devant un pilier, OBS = 1 et MPI = M, comme en E13. En outre, intervient alors le paramètre SIM (simulation) initialisé à 0 en début du programme.
- Après la détermination des deux positions A et B par la méthode habituelle de E13, ces positions sont repérées sur l'écran par des points rouges. On ne calcule pas leur distance au maître, car ils ne sont que les points de départ des deux simulations suivantes, l'une à partir de A (SIM = 1) et l'autre à partir de B (SIM = 2).
- SIM est d'abord incrémenté de 1 et prend donc la valeur 1.
- Le programme mémorise les deux positions A et B et les vitesses d'arrivée correspondantes pour veiller à la condition de non-retour en arrière des mouvements de simulation suivants.
- À partir de A, le programme simule un nouveau pas en prenant pour valeur de départ (PXR, PYR, VXR, VYR) les données ci-dessus relatives au point A. On obtient ainsi une position unique A1 s'il n'y a pas eu de pilier, ou deux positions A1 et A2 s'il y a eu un pilier.
- La fonction d'évaluation calcule alors les distances DISTA1 et DISTA2 ; s'il n'y a pas eu de pilier, on prend une distance fictive DISTA2 = 100 qui sera, par la suite, toujours rejetée.
- SIM est alors incrémenté de 1 et prend la valeur 2.
- Un autre pas est alors simulé dans les mêmes conditions à partir de B et aboutit à une ou deux positions possibles B1 et B2, affectées des distances DISTB1 et DISTB2.
- C'est entre ces quatre distances DISTA1, DISTA2, DISTB1 et DISTB2, dont une ou deux peuvent être fictives, que le programme choisit la plus courte. Si c'est A1 ou A2 qui l'emporte, c'est le point A qui sera occupé par le robot, sinon c'est le point B qui sera choisi.
- Ces opérations sont simples dans leur principe, mais nécessitent tout un arsenal de paramètres et de variables accessoires qu'il faut mémoriser pour les comparaisons, les simulations, et pour ne retenir finalement que les valeurs utiles.
- Les deux, trois ou quatre points finaux A1, A2, B1 et B2 sont visualisés par des points d'interrogation sur l'écran ; de même tous les heurts de pilier, les retours en arrière et les explorations sont jalonnés audiovisuellement comme en E13.

  Le sous-programme heurt pilier comporte par ailleurs une boucle d'attente pour freiner légèrement les opérations et permettre leur suivi.

ÉTAPE E29

DES ALLER ET RETOUR À BLANC

Présentation

Le scénario de cette étape est le même que le précédent mais, comme nous l'avons déjà fait antérieurement, le jalonnement audiovisuel de l'heuristique I a été supprimé et seule subsiste l'action, la marche sûre et sereine de Roméo vers son maître.

On peut constater qu'il n'est guère possible de faire mieux...

Mais le moment est venu, par souci d'honnêteté, de se pencher à nouveau sur les moyens sensoriels mis en œuvre par le robot.

En effet, nous avons toujours pris soin de bien préciser ces moyens dans les programmes précédents, sauf en E13-E14 pour l'heuristique F et en E28-E29 pour la présente heuristique I. C'est que, dans les deux cas, la situation est délicate...

En E13 et E14, on peut admettre à la rigueur que le robot penche son corps vers les deux positions, contiguës à la sienne, susceptibles d'être occupées, et apprécie visuellement les deux distances au maître sans avoir à se déplacer.

Mais en E28 et E29 la chose est impossible, et le seul moyen pour un robot réel consiste à faire véritablement les deux, trois ou quatre déplacements de simulation (simulation réelle, paradoxe des mots...) sur la pointe des pieds, de comparer, en revenant chaque fois sur ses pas.

Lorsque tout ce manège est terminé, il effectue d'un air dégagé le véritable pas. L'expression *y aller par quatre chemins* prend ainsi toute sa saveur.

Curieusement, c'est donc le programme E28 avec son jalonnement, et non pas celui-ci, qui correspondrait au véritable comportement d'un robot "en chair et en os".

Il n'en reste pas moins que l'itinéraire complet, mémorisé par le robot au cours de la première exécution, peut être ensuite brillamment répété sans supercherie suivant la méthode de E20.

Programme

```
100 'ETAPE E29:COMME E28 SANS SONO NI VI
SUALISATION DES SIMULATIONS DE L'HEURIST
```

```
IQUE DE DEGAGEMENT (I)
110 INK 0,22:INK 1,1:INK 2,6:INK 3,14:CL
S:CLEAR
120 DIM C(33,25)'TABLEAU DES COORDONNEES
 DES CASES DE L'ECRAN SUSCEPTIBLES DE RE
CEVOIR UN PILIER
130 SYMBOL AFTER 32:SYMBOL 33,&0,&7C,&D6
,&6C,&38,&FE,&7C,&6C'DESSIN ROBOT;DESSIN
 MAITRE:CODE 250
140 REM NB ALEATOIRES VARIES
150 RANDOMIZE TIME
160 RG=1:SIM=0'INIT RANG DU PARCOURS RG
ET RANG DE LA SIMULATION SIM
170 CLS:H=INT(2*RND):IF H=1 THEN PXM=INT
(25*RND+6):PYM=6 ELSE PXM=6:PYM=INT(12*R
ND+6)'TIRAGE POSITION MAITRE
180 PXR=36:PYR=21:PPXR=PXR:PPYR=PYR:PPXR
T=PXR:PPYRT=PYR'POS DEPART ROBOT,MISE EN
 RESERVE POUR CALCULS DE SIMULATION (PPX
R,PPYR) ET MISE EN RES POUR TRACE
190 PEN 1:LOCATE PXR,PYR:PRINT CHR$(33):
PEN 2:LOCATE PXM,PYM:PRINT CHR$(250)'AFF
 ROBOT ET MAITRE
200 IF RG>1 THEN 280
210 FOR N=0 TO 30:TERR$=INKEY$:IF TERR$<
>" " THEN 220 ELSE 110
220 I=INT((33-PXM)*RND)+PXM-1:J=INT((21-
PYM)/(36-PXM)*(I-PXM))+INT((5*RND-2)*(I-
PXM)/8)+PYM+1
230 IF J>PYM-2 THEN 250
240 J=PYM-1:GOTO 270
250 IF J<>PYM THEN IF J<22 THEN 270 ELSE
 J=21:GOTO 270
260 IF I<>PXM THEN 270 ELSE J=PYM-1
270 PEN 1:LOCATE I,J:PRINT"=":C(I,J)=1:N
EXT'AFF ET REPRESENTATION PILIER
280 SOUND 1,200,80,7:FOR AT=1 TO 2000:NE
XT'TOP DEPART
1000 REM BOUCLE DES MOUVEMENTS
1010 FOR M=1 TO 50'50 MOUVMTS MAXI ADMIS
 POUR ATTEINDRE MAITRE
1020 TERR$=INKEY$:IF TERR$<>" " THEN 103
0 ELSE 110
1030 IF PXR<>PXM THEN NUME=ABS(PYR-PYM):
DENO=ABS(PXR-PXM):IN=NUME/DENO ELSE 1060
'HEUR NAV (1)
1040 IF IN<0.5 THEN VXR=SGN(PXM-PXR):VYR
=0:GOTO 1070
1050 IF IN>2 THEN VXR=0:VYR=SGN(PYM-PYR)
:GOTO 1070
1060 VXR=SGN(PXM-PXR):VYR=SGN(PYM-PYR)
1070 TT=0'INITIALISATION DU RANG TT DES
TENTATIVES POUR TROUVER UNE POSITION LIB
RE APRES HEURT PILIER OU RETOUR EN ARRIE
```

```
RE
1080 PXR=PPXR+VXR:PYR=PPYR+VYR
1090 IF PXR>33 THEN 1610 ELSE GOSUB 3000
'TEST HEURT PILIER
1100 IF OBS=0 THEN 1110 ELSE 1080'RECALC
UL POS
1110 IF RET=0 THEN 1120 ELSE 1080
1120 IF M<>MPI THEN 1360
1130 REM ESSAI AUTRE SENS D'EXPLORATION
1140 IF ESS=1 THEN 1190 ELSE PXR1=PXR:PY
R1=PYR:VXR1=VXR:VYR1=VYR'MEMO 1ERE POS (
A SI SIM=0,A1 SI SIM=1,B1 SI SIM=2)
1150 SSS=SS
1160 IF SSS=1 THEN SS=0 ELSE SS=1'INVERS
ION SENS DE ROTATION
1170 TT=2:ESS=1:D=D1:GOSUB 3250
1180 GOTO 1080'RECALCUL POS
1190 ESS=0:MPI=0'REINIT ESS ET MPI
1200 PXR2=PXR:PYR2=PYR'MEMO 2EME POSITIO
N (B SI SIM=0,A2 SI SIM=1,B2 SI SIM=2)
1210 VXR2=VXR:VYR2=VYR'MEMO VITESSE
1220 REM CALCUL DES 2 DISTANCES DIST1 ET
 DIST2 AU MAITRE
1230 IF SIM=0 THEN 1280'VERS LE DEBUT DE
S SIMULATIONS
1240 HOR1=ABS(PXM-PXR1):VERT1=ABS(PYM-PY
R1)::IF HOR1>VERT1 THEN DIST1=HOR1 ELSE
DIST1=VERT1
1250 HOR2=ABS(PXM-PXR):VERT2=ABS(PYM-PYR
):IF HOR2>VERT2 THEN DIST2=HOR2 ELSE DIS
T2=VERT2
1260 IF SIM=2 THEN 1270 ELSE DISTA1=DIST
1:DISTA2=DIST2:PXRA1=PXR1:PYRA1=PYR1:PXR
A2=PXR2:PYRA2=PYR2:PI1=1:GOTO 1280'CAS S
IM=1:2 POS A1 ET A2 OBT A PARTIR DE A,ME
MO DISTCES ,MEMO POSITS POUR EFFT ULT ET
 SIGNALMT PIL RENC AU COURS 1ERE SIM
1270 DISTB1=DIST1:DISTB2=DIST2:PI2=1:GOT
O 1410'CAS SIM=2:2 POS  OBT A PARTIR DE
B,MEMO DISTCES ET SIGNALMT PIL RENC AU C
OURS 2EME SIM;INUTILE MEMORISER POS PXR1
/PYR1 ET PXR2/PYR2 QUI VONT ETRE UTILISE
ES TELLES QUELLES
1280 SIM=SIM+1:IF SIM=2 THEN 1340
1290 REM SIM=1:MEMO DES 2 POS A ET B D'O
U VONT SE FAIRE LES 2 SIM 1 ET 2 DU PAS
SUIVT DU ROBOT,ET DES VITESSES CORRESPON
DANTES POUR EVITER LES RET ARR
1300 PXRA=PXR1:PYRA=PYR1:VXRA=VXR1:VYRA=
VYR1:PXRB=PXR2:PYRB=PYR2:VXRB=VXR2:VYRB=
VYR2
1310 M=M-2'RETROGRAD M POUR COMPENSER LE
S 2 BOUCLES DE SIMULATION ET EVITER FIN
DE PARCOURS PREMATUREE (M>50)
```

```
1320 REM 1ERE SIM (A PARTIR DE A)
1330 PXR=PXRA:PYR=PYRA:VXR=VXRA:VYR=VYRA
:GOTO 1630
1340 REM 2EME SIM (A PARTIR DE B)
1350 PXR=PXRB:PYR=PYRB:VXR=VXRB:VYR=VYRB
:GOTO 1630
1360 REM PAS DE PILIER
1370 IF SIM=0 THEN 1610 ELSE PXR3=PXR:PY
R3=PYR:HOR3=ABS(PXM-PXR3):VERT3=ABS(PYM-
PYR3)'CAS SIM=1 OU 2
1380 IF HOR3<VERT3 THEN DIST3=VERT3 ELSE
 DIST3=HOR3
1390 IF SIM=1 THEN DISTA1=DIST3:DISTA2=1
00:PXRA3=PXR3:PYRA3=PYR3:PI1=0:GOTO 1280
'POS A1 OBT A PARTIR DE A:MEMO DISTA1,DI
STA2 FICTIVE + ELEVEE,MEMO POS POUR EFFT
 ULT,SIGNALMT PAS DE PIL AU COURS 1ERE S
IM
1400 IF SIM=2 THEN DISTB1=DIST3:DISTB2=1
00:PI2=0'COMME 1390 MAIS INUTILE MEMO PO
S CAR PXR3/PYR3 SERONT UTILISES TELS QUE
LS
1410 REM CHOIX PLUS COURTE DISTANCE AU M
AITRE DES 4 POS A1,A2,B1 ET B2
1420 IF DISTA1<DISTA2 THEN 1470
1430 IF DISTA2<DISTB1 THEN 1460
1440 IF DISTB1<DISTB2 THEN 1530
1450 DIST=DISTB2 :GOTO 1570
1460 IF DISTA2<DISTB2 THEN 1510
1470 IF DISTA1<DISTB1 THEN 1490
1480 GOTO 1440
1490 IF DISTA1<DISTB2 THEN 1520
1500 GOTO 1450
1510 DIST=DISTA2:GOTO 1540
1520 DIST=DISTA1:GOTO 1540
1530 DIST=DISTB1:GOTO 1570
1540 PXR=PXRA:PYR=PYRA
1550 VXR=VXRA:VYR=VYRA
1560 GOTO 1590
1570 PXR=PXRB:PYR=PYRB
1580 VXR=VXRB:VYR=VYRB
1590 REM REINIT SIM
1600 SIM=0
1610 PEN 1:LOCATE 31,1:PRINT M;"":PEN 3:
LOCATE PPXRT,PPYRT:PRINT CHR$(33):SOUND
1,35,5,7:PEN 1:LOCATE PXR,PYR:PRINT CHR$
(33)'AFF TRACE ET ROBOT
1620 PPXRT=PXR:PPYRT=PYR'MEMO POS POUR T
RACE ULT
1630 PPXR=PXR:PPYR=PYR:VVXR=VXR:VVYR=VYR
'MEMO POS ET VIT POUR CALCULS ULT
1640 IF SIM<>0 THEN 1650 ELSE GOSUB 4000
'TEST MAITRE REJOINT (SAUF PENDANT LES S
IMULATIONS)
```

```
1650 NEXT:NEXT M
1660 SOUND 1,500,40,7:RG=RG+1:IF RG=6 TH
EN 110 ELSE PEN 3:LOCATE PXR,PYR:PRINT"C
HR$(33):FOR AT=1 TO 1000:NEXT:GOTO 180'P
ARCOURS NON REUSSI
1670 END
2000 REM ZONE DES SOUS-PROGRAMMES
3000 REM SSPROG HEURT•PILIER ET/OU RETOU
R EN ARRIERE
3010 OBS=0:RET=0
3020 REM TEST PILIER
3030 IF C(PXR,PYR)=1 THEN 3040 ELSE 3050
3040 OBS=1:GOTO 3080
3050 REM TEST RETOUR ARRIERE
3060 IF VXR=-VVXR THEN IF VYR=-VVYR THEN
 3070 ELSE RETURN ELSE RETURN
3070 IF TT>7 THEN RETURN ELSE RET=1
3080 TT=TT+1
3090 MPI=M'MEMO NO BOUCLE
3100 REM ESSAI NOUVELLE DIRECTION:HEURIS
T DEGMT (I)
3110 IF TT>1 THEN 3250'CAS 2EME PASSAGE
ET SUIVANTS
3120 IF VXR=-1 THEN 3150'CAS 1ER PASSAGE
,DETERMINATION DE LA DIRECTION D1 DU 1ER
 HEURT
3130 IF VXR=0 THEN 3180
3140 IF VXR =1 THEN 3200
3150 IF VYR=-1 THEN D1=5:GOTO 3230
3160 IF VYR=0 THEN D1=6:GOTO 3230
3170 IF VYR=1 THEN D1=7:GOTO 3230
3180 IF VYR=-1 THEN D1=4:GOTO 3230
3190 IF VYR=1 THEN D1=8:GOTO 3230
3200 IF VYR=-1 THEN D1=3:GOTO 3230
3210 IF VYR=0 THEN D1=2:GOTO 3230
3220 IF VYR=1 THEN D1=1
3230 D=D1'DIRECTION D DEBUT DE L'EXPLORA
TION
3240 SS=INT(2*RND)'TIRAGE SENS DE L'EXPL
ORATION
3250 REM EXPLORATION POSITION SUIVANTE
3260 IF SS=0 THEN D=D+1:IF D<9 THEN 3280
 ELSE D=D-8:GOTO 3280
3270 D=D-1:IF D>0 THEN 3280 ELSE D=D+8
3280 IF D=1 THEN 3290 ELSE IF D=2 THEN 3
300 ELSE IF D=3 THEN 3310 ELSE IF D=4 TH
EN 3320 ELSE IF D=5 THEN 3330 ELSE IF D=
6 THEN 3340 ELSE IF D=7 THEN 3350 ELSE 3
360
3290 VXR=1:VYR=1:GOTO 3370
3300 VXR=1:VYR=0:GOTO 3370
3310 VXR=1:VYR=-1:GOTO 3370
3320 VXR=0:VYR=-1:GOTO 3370
3330 VXR=-1:VYR=-1:GOTO 3370
```

```
3340 VXR=-1:VYR=0:GOTO 3370
3350 VXR=-1:VYR=1:GOTO 3370
3360 VXR=0:VYR=1
3370 RETURN
4000 REM SSPROG MAITRE REJOINT
4010 IF PXR<>PXM THEN RETURN ELSE IF PYR
<>PYM THEN RETURN
4020 FOR FOIS=1 TO 4:SOUND 1,27,5,7:FOR
AT=1 TO 100:NEXT:NEXT
4030 FOR AT=1 TO 800:NEXT:RG=RG+1:IF RG<
5 THEN PEN 3:LOCATE PPXR,PPYR:PRINT CHR$
(33):GOTO 180 ELSE 110
```

Pas de commentaires techniques.

X

QUELQUES PARTIES DE BRAS DE FER EN PERSPECTIVE

"Le bug est le propre de l'homme [1]."

1. Citation anonyme anticipée à profondeur 2.

Il serait possible de continuer et d'imaginer indéfiniment d'autres comportements pour la traversée de notre labyrinthe, soit en perfectionnant encore ceux qui ont été présentés, soit en imaginant des heuristiques de nature différente.

En effet, comme nous l'avons observé en E6, le concept du couple heuristique navigation-dégagement n'est pas inscrit dans les astres.

Nous avons ainsi pu établir une *heuristique globale* permettant d'atteindre le même but, mais nous laissons au lecteur le plaisir de la découvrir sur les bases suivantes :

- Tout contact direct avec un pilier est interdit au robot ; il ne peut donc pas essayer une position pour vérifier qu'elle est libre ou occupée par un pilier ; il n'a pas le droit de toucher.
- Il dispose par contre d'un sens de la vue lui permettant d'apprécier la direction et la distance de tout pilier visible depuis la case où il se trouve.
- Il peut également, comme dans les exemples précédents, apprécier la direction et la distance de son maître.

Pour en terminer avec les exemples concrets, ce chapitre présente quelques joutes entre robots. Le lecteur pourra même se mêler à l'une d'elles en tant qu'acteur (E32) et insuffler de l'intelligence à son missile lourd dans la dernière (E34) pour tenter de vaincre le propre missile de l'auteur, spécialement armé d'une heuristique de style guerre des étoiles.

Ainsi s'achèvera la valse des robots vidéo, qui nous auront permis d'aborder dans la bonne humeur plusieurs aspects du monde fascinant de l'I.A.

L'intelligence s'y est trouvé placée dans le mince faisceau de quelques projecteurs, mais elle n'a pas perdu pour autant ce qui lui donne tout son prix : elle est par nature incapable de se cerner clairement elle-même, elle est auto-inaccessible ; vouloir comprendre l'intelligence, c'est l'équivalent physique de prétendre se soulever par les cheveux.

*

* *

Un court bilan de toutes ces pérégrinations ne paraît pas inutile, pourvu qu'il reste dans le registre du livre, les pieds sur terre et le nez levé vers le ciel.

On peut louer ou décrier l'Intelligence Artificielle. Mais la somme de réflexion qu'il est nécessaire de mettre en œuvre lors de la réalisation de

ses programmes constitue, qu'on le veuille ou non, une approche pragmatique entièrement nouvelle des comportements humains, même si ces derniers sont souvent schématisés et simplifiés.

Elle ne prétend nullement déboucher sur des pensées philosophiques, elle s'y refuse au contraire car cela serait en contradiction essentielle avec... sa propre philosophie.

Mais il faut savoir lucidement reconnaître que cette approche aura brusquement permis, dans la seconde moitié du XXe siècle, grâce à l'ordinateur simplement considéré comme un manipulateur accéléré de connaissances, d'aboutir à une véritable révélation auparavant totalement inaccessible : celle de deux modes possibles de fonctionnement de la pensée vivante.

Appelons-les, à l'appui des multiples applications déjà réalisées et des quelques exemples proposés dans ce livre, le mode algorithmique et le mode heuristique. Ils ne sont pas ainsi cités dans l'ordre de leurs valeurs supposées, mais dans l'ordre où ils ont émergé dans la conscience humaine, c'est-à-dire aussi dans l'ordre inverse de celui où les hommes les ont inconsciemment pratiqués au cours des âges.

Nous vivons ainsi l'époque privilégiée où un retour aux sources devient possible : heuristique inconsciente, algorithmique inconsciente, puis algorithmique consciente et enfin heuristique consciente, la boucle se ferme.

Ainsi survolés, les algorithmes peuvent apparaître comme des manifestations intelligentes de l'instinct, et les heuristiques comme des manifestations instinctives de l'intelligence, et les deux n'en font plus qu'un.

Les heuristiques sont des raisonnements informels, les algorithmes sont des pulsions formalisées. Ceux-ci disciplinent les errances, celles-là libèrent les fixations.

Les plantes et les animaux ne "pensent" probablement que par heuristiques et les algorithmes semblent spécifiquement humains, car nous étions seuls capables, parmi les espèces, de nous inventer ces chaînes libératrices. En effet, la rigueur artificielle de nos raisonnements constitue un véritable dressage de l'imagination et nous permet curieusement d'accéder, au-delà d'eux, à des heuristiques supérieures.

Les mathématiques et la logique apparaissent à cet égard sous un jour nouveau, comme des outils heuristiques puissants, les tremplins indispensables des envols les plus subtils et les plus riches de la pensée naturelle, et non plus comme d'austères et schématiques instruments incapables de nuances.

L'esprit de géométrie et l'esprit de finesse de Pascal s'y côtoient et s'y fondent, liés par une réciproque émulation, et non plus séparés par l'antagonisme contre nature de l'ère *anteinformatienne*.

À propos des KIPS (*knowledge information processing systems*) qui

seront, espérons-le, les merveilleux ordinateurs dits "de cinquième génération" du grandiose projet japonais, citons ce bref passage du livre déjà évoqué, *La Cinquième Génération* :

"Supposons que, dès notre première rencontre avec un nouveau sujet, on nous présente un nouveau modèle mental bien conçu : une structure solide et vaste que nous pouvons facilement saisir pour commencer, et à laquelle nous pourrons ajouter des détails qui font toute la différence entre la compréhension et l'incompréhension."

Et encore : "La connaissance est un objet digne d'être conçu et fabriqué."

Car les littéraires, les scientifiques, les artistes et les autres, qui ignoreront la nature et les méthodes de l'I.A. et persisteront à les ignorer, se retrouveront assez rapidement naturalisés au fond du poussiéreux musée des *préinformatiens*.

Ainsi que vient de le dire l'un d'eux dans un pathétique constat : "L'Univers m'est déjà devenu étranger."

Mais ne sont-ce pas plutôt eux qui se comportent comme des étrangers à l'Univers d'aujourd'hui ?

Les nouvelles armées de la connaissance vont poursuivre leurs ravages dans les anciennes hordes du faux-semblant ou des suppositions gratuites.

La connaissance et la pensée préparent l'acte, c'est une projection du présent sur l'avenir, une série de tirs à blanc qui précède le coup.

Il est donc aussi légitime, aussi naturel de construire des machines à penser que des machines à agir, déjà entrées depuis des siècles dans nos habitudes. Le plus surprenant est qu'on n'y ait pas songé plus tôt puis que, sachant maintenant les fabriquer, l'on continue à gloser et à faire la fine bouche.

On ne saurait certes oser comparer un homme à une machine ; les machines n'auraient jamais pensé à baptiser les hommes d'un nom aussi péjoratif et aussi porteur de faux jugements : pensez donc, une machine !

Pourtant, les machines sont des machines, avec leur diversité morphologique et fonctionnelle infinie, et les hommes ne sont que des hommes, avec leur spécificité universellement uniforme. Les seconds construisent d'ailleurs les premières pour surmonter leur condition et seraient mal avisés de manifester du mépris pour leurs propres créatures, car ce mépris vise en fait les créateurs ; il est significatif que les contempteurs des machines ignorent ce qu'elles sont et, inversement, que les créateurs de machines n'aient pas ces préventions.

Le dédain d'un homme pour une machine ne recouvrirait-il pas la ran-

cœur banale, par enfant interposé, de quelqu'un qui ignore pour quelqu'un qui connaît ?

Pour les machines à agir, peu rechignent de nos jours car beaucoup connaissent, mais pour les machines à penser l'on craint la concurrence déloyale de ces mystérieux appareils ou plus précisément de leurs propriétaires ; pour ceux-là, le terme de machines est trop beau et donne place aux trucs ou aux bidules. Mais cela relève probablement de la peur confusément ressentie d'une évolution inéluctable qui n'arrange pas tout le monde.

Il y aura demain d'autres philosophes, plus proches de Galilée que d'Aristote, préférant le télescope à la boule de cristal, le Monde aux Nuées, l'ordinateur au crayon. Au total, comme les aurait aimés Montaigne, des hommes à la tête bien faite utilisant la tête bien pleine de petites machines en silice, ni plus ni moins fabuleuses que les fabuleux marteaux-pilons de Zola.

Terminons sur une note actuelle en plaçant dans notre ligne de mire une catégorie exemplaire de charlatans rapidement fossilisables, mais ne sévissant heureusement que dans certains pays exotiques peu développés...

Quelques hommes politiques, dans leurs prestations publiques, appellent irrésistiblement une comparaison avec les ordinateurs, engins dont ils parlent d'ailleurs volontiers.

Ils ne décrivent en effet la réalité qu'à coups de bascules électroniques à deux positions, de bits isolés : oui ou non, pour ou contre, gauche ou droite, gentil ou méchant, généreux ou pingre, pauvre ou riche, travailleur ou profiteur, bien ou mal, nous... ou les autres.

Il leur manque, pour égaler nos petits ordinateurs personnels, non pas hélas ! la parole, mais quelque microprocesseur capable de combiner ces données et d'échafauder les introuvables heuristiques du bonheur universel.

Les micro-informaticiens à tout crin que nous sommes ne sauraient longtemps se laisser prendre à ces sauts de puce, même si les programmes qu'ils bâtissent péniblement sont souvent infestés par de redoutables cohortes de bugs, réfractaires aux insecticides classiques : grammaire, syntaxe, logique, patience, fanatisme...

ÉTAPE E30

COURSE-POURSUITE AU CIRQUE

Présentation

Le but du scénario est de proposer une sorte de piste d'essai permettant de comparer deux à deux des heuristiques. Il s'agit d'une piste carrée autour de laquelle un robot rouge poursuit un robot bleu.

Dans le cas présent, les deux robots utilisent les mêmes heuristiques : 2 et F pour le poursuivant, 5 et F pour le fugitif.

À heuristiques identiques, c'est toujours le poursuivant qui gagne, à cause des limitations du terrain interdisant certaines esquives.

Les poursuites sont agréables à suivre et le lecteur pourra doter les deux partenaires de comportements divers.

Noter que la fonction d'évaluation des distances utilisée ici est différente de celle de E13 afin d'être mieux adaptée à la forme de la piste.

Programme

```
100 'ETAPE E30:PISTE D'ESSAI CARREE DE P
OURSUITE DE 2 ROBOTS;HEURISTIQUES IDENTI
QUES (2)/(F) POUR ROBOT POURSUIVANT ET (
5)/(F) POUR ROBOT POURSUIVI,MODIFICATION
 DE LA FONCTON D'EVALUATION DES DISTANCE
S
110 INK 0,19:INK 1,14:INK 2,6:INK 3,26:C
LS:CLEAR
120 DIM C(40,25)'TABLEAU DES COORDONNEES
 DES CASES DE L'ECRAN SUSCEPTIBLES DE RE
CEVOIR UN OBSTACLE (PILIER OU BORDS DE L
A PISTE)
130 SYMBOL AFTER 32:SYMBOL 33,&0,&7C,&D6
,&6C,&38,&FE,&7C,&6C'DESSIN ROBOTS
140 REM NOMBRES ALEATOIRES VARIES
150 RANDOMIZE TIME
160 CLS
170 REM BORD EXTERIEUR PISTE
180 FOR X=8 TO 32:PEN 1:LOCATE X,3:PRINT
"$":LOCATE X,21:PRINT"$":C(X,3)=1:C(X,21
)=1:NEXT X
190 FOR Y=4 TO 20:LOCATE 8,Y:PRINT"$":LO
CATE 32,Y:PRINT"$":C(8,Y)=1:C(32,Y)=1:NE
XT Y
200 REM BORD INTERIEUR PISTE
```

```
210 FOR X=13 TO 27:LOCATE X,7:PRINT"$":L
OCATE X,17:PRINT"$":C(X,7)=1:C(X,17)=1:N
EXT X
220 FOR Y=8 TO 16:LOCATE 13,Y:PRINT"$":L
OCATE 27,Y:PRINT"$":C(13,Y)=1:C(27,Y)=1:
NEXT Y
230 REM PILIERS SUR PISTE
240 LOCATE 20,4:PRINT"$":LOCATE 20,6:PRI
NT"$":LOCATE 20,18:PRINT"$":LOCATE 20,20
:PRINT"$":LOCATE 9,12:PRINT"$":LOCATE 12
,12:PRINT"$":LOCATE 28,12:PRINT"$":LOCAT
E 31,12:PRINT"$"
250 C(20,4)=1:C(20,6)=1:C(20,18)=1:C(20,
20)=1:C(9,12)=1:C(12,12)=1:C(28,12)=1:C(
31,12)=1
260 PXR1=INT(21*RND)+10:PYR1=5:PXR2=INT(
21*RND)+10:PYR2=19:PEN 1:LOCATE PXR1,PYR
1:PRINT CHR$(33):PEN 2:LOCATE PXR2,PYR2:
PRINT CHR$(33)'POS ET AFF ROBOTS
270 REM MISE EN BOUCLE DES 2 ROBOTS
280 PXA=PXR1:PYA=PYR1:PPXA=PXA:PPYA=PYA:
PXP=PXR2:PYP=PYR2
290 SOUND 1,70,50,7'TOP DEPART
1000 REM BOUCLE DES MOUVEMENTS
1010 FOR M=1 TO 300
1020 TERR$=INKEY$:IF TERR$<>" " THEN 103
0 ELSE 110
1030 REM HEUR NAV (2) POUR POURSUIVANT E
T (5) POUR POURSUIVI
1040 VXA=SGN(PXP-PXA):VYA=SGN(PYP-PYA)
1050 TT=0'INIT RANG TT DES TENTATIVES
1060 IF INT(M/2)<>M/2 THEN 1070 ELSE VXA
=-VXA:VYA=-VYA'CAS DU ROBOT POURSUIVI
1070 PXA=PPXA+VXA:PYA=PPYA+VYA
1080 GOSUB 3000'TEST HEURT PIL ET/OU RET
 ARR
1090 IF OBS=0 THEN 1100 ELSE 1070'RECALC
 POS
1100 IF RET=0 THEN 1110 ELSE 1070
1110 IF M<>MPI THEN 1250
1120 REM ESSAI AUTRE SENS D'EXPLORATION
1130 IF ESS=1 THEN 1170 ELSE PX1=PXA:PY1
=PYA:VX1=VXA:VY1=VYA'MEMO 1ERE POS
1140 SSS=SS:IF SSS=1 THEN SS=0 ELSE SS=1
'INVERSION SENS DE ROTATION
1150 TT=2:ESS=1:D=D1:GOSUB 3250
1160 GOTO 1070'RECALC POS
1170 ESS=0:MPI=0'REINIT ESS ET MPI
1180 REM CHOIX MEILLEURE POS:PLUS COURTE
 DISTANCE POUR POURSUIVANT,PLUS LONGUE P
OUR POURSUIVI
1190 HOR1=ABS(PXP-PX1):VERT1=ABS(PYP-PY1
):DIST1=HOR1+VERT1
1200 HOR2=ABS(PXP-PXA):VERT2=ABS(PYP-PYA
```

```
):DIST2=HOR2+VERT2
1210 IF INT(M/2)<>M/2 THEN 1220 ELSE 123
0
1220 IF DIST1<=DIST2 THEN 1240 ELSE 1250
1230 IF DIST2<=DIST1 THEN 1240 ELSE 1250
1240 PXA=PX1:PYA=PY1:VXA=VX1:VYA=VY1'1ER
E POS CHOISIE
1250 VVXA=VXA:VVYA=VYA'2EME POS CHOISIE
1260 PEN 3:LOCATE PPXA,PPYA:PRINT CHR$(3
3):IF INT(M/2)<>M/2 THEN 1270 ELSE 1280
1270 SOUND 1,40,5,7:PEN 2:LOCATE PXA,PYA
:PRINT CHR$(33):GOTO 1290
1280 SOUND 1,30,5,7:PEN 1:LOCATE PXA,PYA
:PRINT CHR$(33)
1290 PEN 1:LOCATE 35,2:PRINT M;"   "
1300 GOSUB 4000'TEST ROBOT REJOINT
1310 REM INVERSION DES ROBOTS
1320 PXANOUV=PXP:PYANOUV=PYP:PXP=PXA:PYP
=PYA:PXA=PXANOUV:PYA=PYANOUV:PPXA=PXA:PP
YA=PYA:VVXANOUV=VVXP:VVYANOUV=VVYP:VVXP=
VVXA:VVYP=VVYA:VVXA=VVXANOUV:VVYA=VVYANO
UV
1330 NEXT M
1340 END
2000 REM ZONE DES SOUS-PROGRAMMES
3000 REM SSPROG HEURT PILIER ET/OU RET A
RR
3010 OBS=0:RET=0
3020 REM TEST PILIER
3030 IF C(PXA,PYA)=1 THEN 3040 ELSE 3050
3040 OBS=1 :GOTO 3080
3050 REM TEST RET ARR
3060 IF VXA=-VVXA THEN IF VYA=-VVYA THEN
 3070 ELSE RETURN ELSE RETURN
3070 IF TT<8 THEN RET=1:GOTO 3080 ELSE R
ETURN
3080 TT=TT+1:MPI=M'MEMO NO BOUCLE
3090 REM ESSAI NOUV DIRECTION
3100 REM HEUR DEGMT (F):EXPLORATION ORDO
NNEE DES DIRECTIONS EN PARTANT DU PILIER
3110 IF TT>1 THEN 3250'CAS 2EME PASSAGE
ET SUIVANTS
3120 IF VXA=-1 THEN 3150'CAS 1ER PASSAG,
DETERM DE LA DIR D1 DU 1ER HEURT
3130 IF VXA=0 THEN 3180
3140 IF VXA=1 THEN 3200
3150 IF VYA=-1 THEN D1=5:GOTO 3230
3160 IF VYA=0 THEN D1=6:GOTO 3230
3170 IF VYA=1 THEN D1=7:GOTO 3230
3180 IF VYA=-1 THEN D1=4:GOTO 3230
3190 IF VYA=1 THEN D1=8:GOTO 3230
3200 IF VYA=-1 THEN D1=3:GOTO 3230
3210 IF VYA=0 THEN D1=2:GOTO 3230
3220 IF VYA=1 THEN D1=1
```

```
3230 D=D1'DIRECTION D DEBUT DE L'EXPLORA
TION
3240 SS=INT(2*RND)'TIRAGE SENS EXPLORATI
ON
3250 REM EXPLORATION POS SUIVTE
3260 IF SS=0 THEN D=D+1:IF D<9 THEN 3280
 ELSE D=D-8:GOTO 3280
3270 D=D-1:IF D>0 THEN 3280 ELSE D=D+8
3280 IF D=1 THEN 3290 ELSE IF D=2 THEN 3
300 ELSE IF D=3 THEN 3310 ELSE IF D=4 TH
EN 3320 ELSE IF D=5 THEN 3330 ELSE IF D=
6 THEN 3340 ELSE IF D=7 THEN 3350 ELSE 3
360
3290 VXA=1:VYA=1:GOTO 3370
3300 VXA=1:VYA=0:GOTO 3370
3310 VXA=1:VYA=-1:GOTO 3370
3320 VXA=0:VYA=-1:GOTO 3370
3330 VXA=-1:VYA=-1:GOTO 3370
3340 VXA=-1:VYA=0:GOTO 3370
3350 VXA=-1:VYA=1:GOTO 3370
3360 VXA=0:VYA=1
3370 RETURN
4000 REM SSPROG ROBOT REJOINT
4010 IF PXA<>PXP THEN RETURN ELSE IF PYA
<>PYP THEN RETURN
4020 SOUND 1,1000,20,7:SOUND 1,400,30,5:
SOUND 1,2000,50,7:FOR AT=1 TO 2000:NEXT
4030 GOTO 110
```

Commentaires techniques

- Il y a une seule boucle des mouvements pour les deux robots, suivant la méthode maintenant classique de E16. Mais cela n'est possible que si les deux robots utilisent les mêmes heuristiques.
- La fonction d'évaluation des distances DIST1 et DIST2 s'écrit cette fois :

  DIST = HOR + VERT.

  Ce n'est ni la distance de E13, ni la distance géométrique de Pythagore. C'est une autre distance heuristique qui évite au robot rouge, lorsqu'il est sur le côté opposé de la piste, de s'éloigner du robot bleu comme c'est parfois le cas si l'on adopte la fonction de E13. Le lecteur pourra chercher pourquoi.
- Bien entendu, le robot rouge choisit la distance la plus courte et le fugitif la distance la plus longue.
- Il est probable que l'obligation de non-retour en arrière est néfaste au robot poursuivi ; par ailleurs, il doit être possible d'améliorer encore

la fonction d'évaluation : il n'est pas logique que le fugitif longe le bord extérieur de la piste car il maintiendrait mieux son avance en prenant au plus court, le long du bord intérieur.

- Ne pas oublier, toutefois, que les heuristiques et la fonction d'évaluation doivent rester les mêmes pour les deux robots, sinon ce serait trop facile.

ÉTAPE E31

UNE PISTE PEU CARROSSABLE

Présentation

Les deux robots se poursuivent sur la même piste qu'en E30, mais de nouveaux obstacles, tirés au hasard, surgissent constamment sur leur chemin.

L'utilisation des prédicats permet de représenter cet environnement variable d'une manière souple et facile.

Programme

```
100 'ETAPE E31:COMME E30 AVEC MODIFICATI
ON DE L'ENVIRONNEMENT PENDANT L'EXECUTIO
N DU PROGRAMME:INTRODUCTION D'OBSTACLES
SUPPLEMENTAIRES AU COURS DE LA POURSUITE
110 INK 0,19:INK 1,14:INK 2,6:INK 3,26:C
LS:CLEAR
120 DIM C(40,25)'TABLEAU DES COORDONNEES
 DES CASES DE L'ECRAN SUSCEPTIBLES DE RE
CEVOIR UN OBSTACLE (PILIER OU BORDS DE L
A PISTE)
130 SYMBOL AFTER 32:SYMBOL 33,&0,&7C,&D6
,&6C,&38,&FE,&7C,&6C'DESSIN ROBOTS
140 REM NOMBRES ALEATOIRES VARIES
150 RANDOMIZE TIME
160 CLS
170 REM BORD EXTERIEUR PISTE
180 FOR X=8 TO 32:PEN 1:LOCATE X,3:PRINT
"$":LOCATE X,21:PRINT"$":C(X,3)=1:C(X,21
)=1:NEXT X
190 FOR Y=4 TO 20:LOCATE 8,Y:PRINT"$":LO
CATE 32,Y:PRINT"$":C(8,Y)=1:C(32,Y)=1:NE
XT Y
200 REM BORD INTERIEUR PISTE
210 FOR X=13 TO 27:LOCATE X,7:PRINT"$":L
OCATE X,17:PRINT"$":C(X,7)=1:C(X,17)=1:N
EXT X
220 FOR Y=8 TO 16:LOCATE 13,Y:PRINT"$":L
OCATE 27,Y:PRINT"$":C(13,Y)=1:C(27,Y)=1:
NEXT Y
230 REM PILIERS SUR PISTE
240 LOCATE 20,4:PRINT"$":LOCATE 20,6:PRI
NT"$":LOCATE 20,18:PRINT"$":LOCATE 20,20
```

```
:PRINT"$":LOCATE 9,12:PRINT"$":LOCATE 12
,12:PRINT"$":LOCATE 28,12:PRINT"$":LOCAT
E 31,12:PRINT"$"
250 C(20,4)=1:C(20,6)=1:C(20,18)=1:C(20,
20)=1:C(9,12)=1:C(12,12)=1:C(28,12)=1:C(
31,12)=1
260 PXR1=INT(21*RND)+10:PYR1=5:PXR2=INT(
21*RND)+10:PYR2=19:PEN 1:LOCATE PXR1,PYR
1:PRINT CHR$(33):PEN 2:LOCATE PXR2,PYR2:
PRINT CHR$(33)'POS ET AFF ROBOTS
270 REM MISE EN BOUCLE DES 2 ROBOTS
280 PXA=PXR1:PYA=PYR1:PPXA=PXA:PPYA=PYA:
PXP=PXR2:PYP=PYR2
290 SOUND 1,70,50,7'TOP DEPART
1000 REM BOUCLE DES MOUVEMENTS
1010 FOR M=1 TO 300:TERR$=INKEY$:IF TERR
$=" " THEN 110'CHANGMT DE TERRAIN
1020 GOSUB 5000'INTRODUCTION D'OBSTACLES
 SUPPLEMENTAIRES
1030 REM HEUR NAV (2) POUR POURSUIVANT E
T (5) POUR POURSUIVI
1040 VXA=SGN(PXP-PXA):VYA=SGN(PYP-PYA)
1050 TT=0'INIT RANG TT DES TENTATIVES
1060 IF INT(M/2)<>M/2 THEN 1070 ELSE VXA
=-VXA:VYA=-VYA'CAS DU ROBOT POURSUIVI
1070 PXA=PPXA+VXA:PYA=PPYA+VYA
1080 GOSUB 3000'TEST HEURT PIL ET/OU RET
 ARR
1090 IF OBS=0 THEN 1100 ELSE 1070'RECALC
 POS
1100 IF RET=0 THEN 1110 ELSE 1070
1110 IF M<>MPI THEN 1250
1120 REM ESSAI AUTRE SENS D'EXPLORATION
1130 IF ESS=1 THEN 1170 ELSE PX1=PXA:PY1
=PYA:VX1=VXA:VY1=VYA'MEMO 1ERE POS
1140 SSS=SS:IF SSS=1 THEN SS=0 ELSE SS=1
'INVERSION SENS DE ROTATION
1150 TT=2:ESS=1:D=D1:GOSUB 3250
1160 GOTO 1070'RECALC POS
1170 ESS=0:MPI=0'REINIT ESS ET MPI
1180 REM CHOIX MEILLEURE POS:PLUS COURTE
 DISTANCE POUR POURSUIVANT,PLUS LONGUE P
OUR POURSUIVI
1190 HOR1=ABS(PXP-PX1):VERT1=ABS(PYP-PY1
):DIST1=HOR1+VERT1
1200 HOR2=ABS(PXP-PXA):VERT2=ABS(PYP-PYA
):DIST2=HOR2+VERT2
1210 IF INT(M/2)<>M/2 THEN 1220 ELSE 123
0
1220 IF DIST1<=DIST2 THEN 1240 ELSE 1250
1230 IF DIST2<=DIST1 THEN 1240 ELSE 1250
1240 PXA=PX1:PYA=PY1:VXA=VX1:VYA=VY1'1ER
E POS CHOISIE
1250 VVXA=VXA:VVYA=VYA'2EME POS CHOISIE
```

```
1260 PEN 3:LOCATE PPXA,PPYA:PRINT CHR$(3
3):IF INT(M/2)<>M/2 THEN 1270 ELSE 1280
1270 SOUND 1,40,5,7:PEN 2:LOCATE PXA,PYA
:PRINT CHR$(33):GOTO 1290
1280 SOUND 1,30,5,7:PEN 1:LOCATE PXA,PYA
:PRINT CHR$(33)
1290 PEN 1:LOCATE 35,2:PRINT M;"  "
1300 GOSUB 4000'TEST ROBOT REJOINT
1310 REM INVERSION DES ROBOTS
1320 PXANOUV=PXP:PYANOUV=PYP:PXP=PXA:PYP
=PYA:PXA=PXANOUV:PYA=PYANOUV:PPXA=PXA:PP
YA=PYA:VVXANOUV=VVXP:VVYANOUV=VVYP:VVXP=
VVXA:VVYP=VVYA:VVXA=VVXANOUV:VVYA=VVYANO
UV
1330 NEXT M
1340 END
2000 REM ZONE DES SOUS-PROGRAMMES
3000 REM SSPROG HEURT PILIER ET/OU RET A
RR
3010 OBS=0:RET=0
3020 REM TEST PILIER
3030 IF C(PXA,PYA)=1 THEN 3040 ELSE 3050
3040 OBS=1 :GOTO 3080
3050 REM TEST RET ARR
3060 IF VXA=-VVXA THEN IF VYA=-VVYA THEN
 3070 ELSE RETURN ELSE RETURN
3070 IF TT<8 THEN RET=1:GOTO 3080 ELSE R
ETURN
3080 TT=TT+1:MPI=M'MEMO NO BOUCLE
3090 REM ESSAI NOUV DIRECTION
3100 REM HEUR DEGMT (F):EXPLORATION ORDO
NNEE DES DIRECTIONS EN PARTANT DU PILIER
3110 IF TT>1 THEN 3250'CAS 2EME PASSAGE
ET SUIVANTS
3120 IF VXA=-1 THEN 3150'CAS 1ER PASSAG,
DETERM DE LA DIR D1 DU 1ER HEURT
3130 IF VXA=0 THEN 3180
3140 IF VXA=1 THEN 3200
3150 IF VYA=-1 THEN D1=5:GOTO 3230
3160 IF VYA=0 THEN D1=6:GOTO 3230
3170 IF VYA=1 THEN D1=7:GOTO 3230
3180 IF VYA=-1 THEN D1=4:GOTO 3230
3190 IF VYA=1 THEN D1=8:GOTO 3230
3200 IF VYA=-1 THEN D1=3:GOTO 3230
3210 IF VYA=0 THEN D1=2:GOTO 3230
3220 IF VYA=1 THEN D1=1
3230 D=D1'DIRECTION D DEBUT DE L'EXPLORA
TION
3240 SS=INT(2*RND)'TIRAGE SENS EXPLORATI
ON
3250 REM EXPLORATION POS SUIVTE
3260 IF SS=0 THEN D=D+1:IF D<9 THEN 3280
 ELSE D=D-8:GOTO 3280
3270 D=D-1:IF D>0 THEN 3280 ELSE D=D+8
```

```
3280 IF D=1 THEN 3290 ELSE IF D=2 THEN 3
300 ELSE IF D=3 THEN 3310 ELSE IF D=4 TH
EN 3320 ELSE IF D=5 THEN 3330 ELSE IF D=
6 THEN 3340 ELSE IF D=7 THEN 3350 ELSE 3
360
3290 VXA=1:VYA=1:GOTO 3370
3300 VXA=1:VYA=0:GOTO 3370
3310 VXA=1:VYA=-1:GOTO 3370
3320 VXA=0:VYA=-1:GOTO 3370
3330 VXA=-1:VYA=-1:GOTO 3370
3340 VXA=-1:VYA=0:GOTO 3370
3350 VXA=-1:VYA=1:GOTO 3370
3360 VXA=0:VYA=1
3370 RETURN
4000 REM SSPROG ROBOT REJOINT
4010 IF PXA<>PXP THEN RETURN ELSE IF PYA
<>PYP THEN RETURN
4020 SOUND 1,1000,20,7:SOUND 1,400,30,5:
SOUND 1,2000,50,7:FOR AT=1 TO 2000:NEXT
4030 GOTO 110
5000 REM SSPROG INTRODUCTION PROGRESSIVE
 D'OBSTACLES SUPPLEMENTAIRES
5010 IF M-4*INT(M/4)=0 THEN PXO=INT(15*R
ND)+13:PYO=INT(3*RND)+4:C(PXO,PYO)=1:PEN
 1:LOCATE PXO,PYO:PRINT"$":RETURN
5020 IF M-4*INT(M/4)=1 THEN PXO=INT(15*R
ND)+13:PYO=INT(3*RND)+18:C(PXO,PYO)=1:PE
N 1:LOCATE PXO,PYO:PRINT"$":RETURN
5030 IF M-4*INT(M/4)=2 THEN PXO=INT(4*RN
D)+9:PYO=INT(11*RND)+7:C(PXO,PYO)=1:PEN
1:LOCATE PXO,PYO:PRINT"$":RETURN
5040 PXO=INT(4*RND)+28:PYO=INT(11*RND)+7
:C(PXO,PYO)=1:PEN 1:LOCATE PXO,PYO:PRINT
"$":RETURN
```

Commentaires techniques

- Un obstacle supplémentaire est introduit à chaque boucle nouvelle par renvoi à un sous-programme spécial.
- Les obstacles sont successivement placés dans les parties haute, basse, gauche et droite de la piste.

ÉTAPE E32

SUS AU MÉCHANT!

Présentation

Il s'agit encore de la piste et des deux robots de E30-E31.

Mais cette fois, l'opérateur pose lui-même des obstacles supplémentaires pour retarder le moment où le robot rouge rattrapera sa victime. Ces obstacles sont mis en place au clavier à l'aide des quatre touches de déplacement du curseur, puis validés par appui sur la touche Entrée.

Le joueur n'a pas le droit d'obstruer totalement la piste, pour laisser leur chance aux deux protagonistes.

Là encore les prédicats permettent une programmation assez facile.

Programme

```
100 'ETAPE E32:JEU D'ALLIANCE AVEC LE RO
BOT POURSUIVI DE E30;MISE EN PLACE D'OBS
TACLES POUR RETARDER LA PROGRESSION DU R
OBOT POURSUIVANT,A L'AIDE DES TOUCHES-CU
RSEUR ET DE LA TOUCHE ENTER
110 INK 0,19:INK 1,14:INK 2,6:INK 3,26:C
LS:CLEAR
120 DIM C(40,25)'TABLEAU DES COORDONNEES
 DES CASES DE L'ECRAN SUSCEPTIBLES DE RE
CEVOIR UN OBSTACLE (PILIER OU BORDS DE L
A PISTE)
130 SYMBOL AFTER 32:SYMBOL 33,&0,&7C,&D6
,&6C,&38,&FE,&7C,&6C'DESSIN ROBOTS
140 REM NOMBRES ALEATOIRES VARIES
150 RANDOMIZE TIME
160 CLS
170 REM BORD EXTERIEUR PISTE
180 FOR X=8 TO 32:PEN 1:LOCATE X,3:PRINT
"$":LOCATE X,21:PRINT"$":C(X,3)=1:C(X,21
)=1:NEXT X
190 FOR Y=4 TO 20:LOCATE 8,Y:PRINT"$":LO
CATE 32,Y:PRINT"$":C(8,Y)=1:C(32,Y)=1:NE
XT Y
200 REM BORD INTERIEUR PISTE
210 FOR X=13 TO 27:LOCATE X,7:PRINT"$":L
OCATE X,17:PRINT"$":C(X,7)=1:C(X,17)=1:N
EXT X
220 FOR Y=8 TO 16:LOCATE 13,Y:PRINT"$":L
```

```
OCATE 27,Y:PRINT"$":C(13,Y)=1:C(27,Y)=1:
NEXT Y
230 REM PILIERS SUR PISTE
240 LOCATE 20,4:PRINT"$":LOCATE 20,6:PRI
NT"$":LOCATE 20,18:PRINT"$":LOCATE 20,20
:PRINT"$":LOCATE 9,12:PRINT"$":LOCATE 12
,12:PRINT"$":LOCATE 28,12:PRINT"$":LOCAT
E 31,12:PRINT"$"
250 C(20,4)=1:C(20,6)=1:C(20,18)=1:C(20,
20)=1:C(9,12)=1:C(12,12)=1:C(28,12)=1:C(
31,12)=1
260 PXR1=INT(21*RND)+10:PYR1=5:PXR2=INT(
21*RND)+10:PYR2=19:PEN 1:LOCATE PXR1,PYR
1:PRINT CHR$(33):PEN 2:LOCATE PXR2,PYR2:
PRINT CHR$(33)'POS ET AFF ROBOTS
270 REM MISE EN BOUCLE DES 2 ROBOTS
280 PXA=PXR1:PYA=PYR1:PPXA=PXA:PPYA=PYA:
PXP=PXR2:PYP=PYR2
290 SOUND 1,70,50,7'TOP DEPART
300 REM POS ET AFF JOUEUR
310 PXJ=21:PYJ=6:PEN 3:LOCATE PXJ,PYJ:PR
INT"^":PPXJ=PXJ:PPYJ=PYJ
1000 REM BOUCLE DES MOUVEMENTS
1010 FOR M=1 TO 500
1020 TERR$=INKEY$:IF TERR$=" " THEN 110
1030 REM HEUR NAV (2) POUR POURSUIVANT E
T (5) POUR POURSUIVI
1040 VXA=SGN(PXP-PXA):VYA=SGN(PYP-PYA)
1050 TT=0'INIT RANG TT DES TENTATIVES
1060 IF INT(M/2)<>M/2 THEN 1070 ELSE VXA
=-VXA:VYA=-VYA'CAS DU ROBOT POURSUIVI
1070 PXA=PPXA+VXA:PYA=PPYA+VYA
1080 GOSUB 3000'TEST HEURT PIL ET/OU RET
 ARR
1090 IF OBS=0 THEN 1100 ELSE 1070'RECALC
 POS
1100 IF RET=0 THEN 1110 ELSE 1070
1110 IF M<>MPI THEN 1250
1120 REM ESSAI AUTRE SENS D'EXPLORATION
1130 IF ESS=1 THEN 1170 ELSE PX1=PXA:PY1
=PYA:VX1=VXA:VY1=VYA'MEMO 1ERE POS
1140 SSS=SS:IF SSS=1 THEN SS=0 ELSE SS=1
'INVERSION SENS DE ROTATION
1150 TT=2:ESS=1:D=D1:GOSUB 3250
1160 GOTO 1070'RECALC POS
1170 ESS=0:MPI=0'REINIT ESS ET MPI
1180 REM CHOIX MEILLEURE POS:PLUS COURTE
 DISTANCE POUR POURSUIVANT,PLUS LONGUE P
OUR POURSUIVI
1190 HOR1=ABS(PXP-PX1):VERT1=ABS(PYP-PY1
):DIST1=HOR1+VERT1
1200 HOR2=ABS(PXP-PXA):VERT2=ABS(PYP-PYA
):DIST2=HOR2+VERT2
1210 IF INT(M/2)<>M/2 THEN 1220 ELSE 123
```

```
0
1220 IF DIST1<=DIST2 THEN 1240 ELSE 1250
1230 IF DIST2<=DIST1 THEN 1240 ELSE 1250
1240 PXA=PX1:PYA=PY1:VXA=VX1:VYA=VY1'1ER
E POS CHOISIE
1250 VVXA=VXA:VVYA=VYA'2EME POS CHOISIE
1260 PEN 3:LOCATE PPXA,PPYA:PRINT CHR$(3
3):IF INT(M/2)<>M/2 THEN 1270 ELSE 1280
1270 SOUND 1,40,5,7:PEN 2:LOCATE PXA,PYA
:PRINT CHR$(33):GOTO 1290
1280 SOUND 1,30,5,7:PEN 1:LOCATE PXA,PYA
:PRINT CHR$(33)
1290 PEN 1:LOCATE 35,2:PRINT M;"   "
1300 GOSUB 4000'TEST ROBOT REJOINT
1310 REM INVERSION DES ROBOTS
1320 PXANOUV=PXP:PYANOUV=PYP:PXP=PXA:PYP
=PYA:PXA=PXANOUV:PYA=PYANOUV:PPXA=PXA:PP
YA=PYA:VVXANOUV=VVXP:VVYANOUV=VVYP:VVXP=
VVXA:VVYP=VVYA:VVXA=VVXANOUV:VVYA=VVYANO
UV
1330 REM COMMANDE DEPLACEMENT JOUEUR ET
INTROD OBSTACLES
1340 GOSUB 5000
1350 FOR AT=1 TO 100:NEXT:NEXT M
1360 GOTO 100'RENVOI A NOUVELLE PARTIE
2000 REM ZONE DES SOUS-PROGRAMMES
3000 REM SSPROG HEURT PILIER ET/OU RET A
RR
3010 OBS=0:RET=0
3020 REM TEST PILIER
3030 IF C(PXA,PYA)=1 THEN 3040 ELSE 3050
3040 OBS=1 :GOTO 3080
3050 REM TEST RET ARR
3060 IF VXA=-VVXA THEN IF VYA=-VVYA THEN
 3070 ELSE RETURN ELSE RETURN
3070 IF TT<8 THEN RET=1:GOTO 3080 ELSE R
ETURN
3080 TT=TT+1:MPI=M'MEMO NO BOUCLE
3090 REM ESSAI NOUV DIRECTION
3100 REM HEUR DEGMT (F):EXPLORATION ORDO
NNEE DES DIRECTIONS EN PARTANT DU PILIER
3110 IF TT>1 THEN 3250'CAS 2EME PASSAGE
ET SUIVANTS
3120 IF VXA=-1 THEN 3150'CAS 1ER PASSAG,
DETERM DE LA DIR D1 DU 1ER HEURT
3130 IF VXA=0 THEN 3180
3140 IF VXA=1 THEN 3200
3150 IF VYA=-1 THEN D1=5:GOTO 3230
3160 IF VYA=0 THEN D1=6:GOTO 3230
3170 IF VYA=1 THEN D1=7:GOTO 3230
3180 IF VYA=-1 THEN D1=4:GOTO 3230
3190 IF VYA=1 THEN D1=8:GOTO 3230
3200 IF VYA=-1 THEN D1=3:GOTO 3230
3210 IF VYA=0 THEN D1=2:GOTO 3230
```

```
3220 IF VYA=1 THEN D1=1
3230 D=D1'DIRECTION D DEBUT DE L'EXPLORA
TION
3240 SS=INT(2*RND)'TIRAGE SENS EXPLORATI
ON
3250 REM EXPLORATION POS SUIVTE
3260 IF SS=0 THEN D=D+1:IF D<9 THEN 3280
 ELSE D=D-8:GOTO 3280
3270 D=D-1:IF D>0 THEN 3280 ELSE D=D+8
3280 IF D=1 THEN 3290 ELSE IF D=2 THEN 3
300 ELSE IF D=3 THEN 3310 ELSE IF D=4 TH
EN 3320 ELSE IF D=5 THEN 3330 ELSE IF D=
6 THEN 3340 ELSE IF D=7 THEN 3350 ELSE 3
360
3290 VXA=1:VYA=1:GOTO 3370
3300 VXA=1:VYA=0:GOTO 3370
3310 VXA=1:VYA=-1:GOTO 3370
3320 VXA=0:VYA=-1:GOTO 3370
3330 VXA=-1:VYA=-1:GOTO 3370
3340 VXA=-1:VYA=0:GOTO 3370
3350 VXA=-1:VYA=1:GOTO 3370
3360 VXA=0:VYA=1
3370 RETURN
4000 REM SSPROG ROBOT REJOINT
4010 IF PXA<>PXP THEN RETURN ELSE IF PYA
<>PYP THEN RETURN
4020 SOUND 1,1000,20,7:SOUND 1,400,30,5:
SOUND 1,2000,50,7:FOR AT=1 TO 2000:NEXT
4030 GOTO 110
4040 REM SSPROG INTRODUCTION PROGRESSIVE
 D'OBSTACLES SUPPLEMENTAIRES
4050 IF M-4*INT(M/4)=0 THEN PXO=INT(15*R
ND)+13:PYO=INT(3*RND)+4:C(PXO,PYO)=1:PEN
 1:LOCATE PXO,PYO:PRINT"$":RETURN
4060 IF M-4*INT(M/4)=1 THEN PXO=INT(15*R
ND)+13:PYO=INT(3*RND)+18:C(PXO,PYO)=1:PE
N 1:LOCATE PXO,PYO:PRINT"$":RETURN
4070 IF M-4*INT(M/4)=2 THEN PXO=INT(4*RN
D)+9:PYO=INT(11*RND)+7:C(PXO,PYO)=1:PEN
1:LOCATE PXO,PYO:PRINT"$":RETURN
4080 PXO=INT(4*RND)+28:PYO=INT(11*RND)+7
:C(PXO,PYO)=1:PEN 1:LOCATE PXO,PYO:PRINT
"$":RETURN
5000 REM SSPROG DEPLACEMENT JOUEUR ET IN
TRODUCTION OBSTACLES
5010 J$=INKEY$:IF J$=CHR$(242) THEN VXJ=
-1:VYJ=0:GOTO 5060
5020 IF J$=CHR$(243) THEN VXJ=1:VYJ=0:GO
TO 5060
5030 IF J$=CHR$(241) THEN VXJ=0:VYJ=1:GO
TO 5060
5040 IF J$=CHR$(240) THEN VXJ=0:VYJ=-1:G
OTO 5060
5050 VXJ=0:VYJ=0:GOTO 5070'CAS OU AUCUNE
```

```
 TOUCHE APPUYEE
5060 SOUND 1,100,10,7
5070 PXJ=PXJ+VXJ:PYJ=PYJ+VYJ
5080 REM EFFETS DE BORDS
5090 IF PXJ>8 THEN 5100 ELSE PXJ=9:GOTO
5130
5100 IF PXJ<32 THEN 5110 ELSE PXJ=31
5110 IF PYJ>3 THEN 5120 ELSE PYJ=4:GOTO
5130
5120 IF PYJ<21 THEN 5130 ELSE PYJ=20
5130 IF C(PPXJ,PPYJ)=1 THEN PEN 1:LOCATE
 PPXJ,PPYJ:PRINT"$" ELSE LOCATE PPXJ,PPY
J:PRINT" "
5140 IF C(PXJ,PYJ)=1 THEN PEN 1:LOCATE P
XJ,PYJ:PRINT"$" ELSE PEN 3:LOCATE PXJ,PY
J:PRINT"^"
5150 PPXJ=PXJ:PPYJ=PYJ
5160 IF J$<>CHR$(224) THEN RETURN ELSE S
OUND 1,20,5,7:PEN 1:LOCATE PXJ,PYJ:PRINT
"$":C(PXJ,PYJ)=1:RETURN
```

Commentaires techniques

- La commande de déplacement du joueur, symbolisée sur l'écran par la flèche verticale de code 94, est programmée vers la fin de la boucle des mouvements par renvoi à un sous-programme spécial.
- Ce sous-programme surveille le clavier par une instruction INKEY$, réponse A$.

ÉTAPE E33

BANC D'ESSAI POUR HEURISTIQUES CONCURRENTES

Présentation

Nous avons vu en E4 que certaines heuristiques, tout en conduisant à des nombres de pas identiques pour un parcours donné, doivent être cependant considérées comme supérieures à d'autres, car leur cadence de réflexion, donc d'exécution, est plus rapide.

C'est le cas de l'heuristique de navigation 2 par rapport à l'heuristique 1 : 2 est plus simple et plus rapidement calculée, sa boucle est parcourue plus vite, la cadence des pas est plus élevée.

Le programme E4 permet de distinguer nettement la différence. Celui de la présente étape est plus parlant, car on y voit deux robots, dotés chacun de l'une des deux heuristiques (robot rouge : 1 ; robot bleu : 2) avancer ensemble vers le but.

Le programme propose pour un trajet donné, soit de comparer les nombres de pas (option P), soit de comparer les temps de parcours (option T).

Chaque robot possède sa propre boucle des mouvements.

Dans l'option P, les robots font un pas chacun à leur tour. Les deux boucles sont parcourues successivement, en série. De ce fait, bien que la boucle 2 soit parcourue plus rapidement que la boucle 1, la cadence générale est forcément la même et seul le nombre de pas fait la différence entre les deux heuristiques.

En l'occurrence, les deux robots arrivent ensemble au but.

Par contre, dans l'option T, chaque robot effectue à son tour une *opération mentale élémentaire*, c'est-à-dire parcourt un morceau de sa boucle, et c'est ensuite le tour de l'autre robot.

Les deux boucles ont été ainsi découpées en tranches dont les temps d'exécution sont estimés sensiblement égaux, la boucle 1 comportant donc un nombre de tranches plus élevé que la boucle 2.

Le programme parcourt alternativement une tranche de chaque boucle, en parallèle, et restitue donc à l'écran l'échelle des temps réels d'exécution.

Quel que soit le parcours, le robot bleu 2 arrive le premier : l'heuristique 2 est supérieure à l'heuristique 1.

Programme

```
100 'ETAPE E33:STRUCTURE-TYPE D'UN PROGR
AMME DE COMPARAISON DES "INTELLIGENCES"
DE 2 HEURIST VIS A VIS D'UNE EPREUVE (SI
TUATION ET OBJECTIF) DONNEE
110 'DEF EPREUVE:-SITUATION:2 ROBOTS,POS
 TIREE AU HASARD -OBJECTIF:REJ MAITRE,PO
S EGALMT TIREE AU HASARD,DANS LE TEMPS M
INIMUM (OPTION T) OU AVEC LE NOMBRE MINI
MUM DE PAS (OPTION P);POSSIB CHANGER D'O
PTION PAR TOUCHE D'ESPACEMENT
120 INK 0,22:INK 1,1:INK 2,6:CLS:CLEAR
130 SYMBOL AFTER 32:SYMBOL 33,&0,&7C,&D6
,&6C,&38,&FE,&7C,&6C
140 PEN 1:LOCATE 4,10:PRINT"TEMPS MINI O
U PAS MINI (T OU P)?"
150 RANDOMIZE TIME:OP$=INKEY$:IF OP$<>"T
" AND OP$<>"P" THEN SOUND 1,25,5,7:FOR A
T=1 TO 100:NEXT:GOTO 150
160 CLS:PXR1=39:PYR1=INT(21*RND)+2:PXR2=
PXR1:PYR2=PYR1'POSITION DEPART ROBOTS
170 PXM=1:PYM=INT(21*RND)+2:PEN 1: LOCAT
E PXM,PYM:PRINT CHR$(248)
180 FOR FOIS=1 TO 10:PEN 1:LOCATE PXR1,P
YR1:PRINT CHR$(33):SOUND 1,30,2,7:PEN 2:
LOCATE PXR2,PYR2:PRINT CHR$(33):FOR AT=1
 TO 50:NEXT:NEXT
190 SOUND 1,100,40,7:N=1
1000 REM BOUCLE 1:MOUVEMENTS ROBOT NO 1,
HEURIST NAV (1)
1010 FOR AT=1 TO 50:NEXT:IF PXM-PXR1=0 T
HEN VXR1=0:VYR1=SGN(PYM-PYR1):GOTO 1040
ELSE NUME=ABS(PYM-PYR1):DENO=ABS(PXM-PXR
1):IN=NUME/DENO
1020 GOSUB 5000
1030 IF IN<0.5 THEN VXR1=SGN(PXM-PXR1):V
YR1=0 ELSE IF IN>2 THEN VXR1=0:VYR1=SGN(
PYM-PYR1) ELSE VXR1=SGN(PXM-PXR1):VYR1=S
GN(PYM-PYR1)
1040 GOSUB 5000
1050 PXR1=PXR1+VXR1:PYR1=PYR1+VYR1
     :GOSUB 5000
1060 PEN 1:LOCATE PXR1,PYR1:PRINT CHR$(3
3):SOUND 1,80,5,7
          :GOSUB 5000
1070 IF OP$="T" THEN 1000 ELSE 2000
2000 REM BOUCLE 2:MOUVEMENTS ROBOT NO 2,
HEURIST NAV (2)
2010 FOR AT=1 TO 50:NEXT:N=2:VXR2=SGN(PX
M-PXR2):VYR2=SGN(PYM-PYR2)
                              :IF OP$="T" THE
N RETURN ELSE 2020
2020 N=3:PXR2=PXR2+VXR2:PYR2=PYR2+VYR2
```

```
     :IF OP$="T" THEN RETURN ELSE 2030
2030 N=4:PEN 2:LOCATE PXR2,PYR2:PRINT CH
R$(33):SOUND 1,70,5,7
     :IF OP$="T" THEN RETURN ELSE 2040

2040 GOSUB 4000'TEST MAITRE REJOINT
2050 IF OP$="T" THEN 2000 ELSE 1000
2060 END
3000 REM ZONE DES SOUS-PROGRAMMES
4000 REM SSPROG MAITRE REJOINT
4010 IF PXR1=PXM AND PYR1=PYM THEN 4020
ELSE IF PXR2=PXM AND PYR2=PYM THEN 4020
ELSE RETURN
4020 SOUND 1,40,50,7:FOR AT=1 TO 2000:NE
XT:TERR$=INKEY$:IF TERR$=" " THEN 120 EL
SE 160
4040 REM BOUCLE 1:MOUVMTS ROBOT NO 1,HEU
R NAV (1)
5000 REM SSPROG TRANSFERT OPTIONNEL DE B
OUCLE 1 A BOUCLE 2
5010 IF OP$="T" THEN ON N GOSUB 2010,202
0,2030,2040:RETURN ELSE RETURN
```

Commentaires techniques

- L'option T ou P est introduite au clavier à l'aide d'une instruction INKEY$ en tête de programme : réponse par variable OP$ (option).
- Les positions de départ et d'arrivée, tirées au hasard, sont bien entendu identiques pour les deux robots.
- Le programme comporte deux boucles, l'une pour le robot rouge, l'autre pour le bleu.
- Chacune est découpée en tranches repérables, dans la boucle 1 par le renvoi GOSUB 5000, et dans la boucle 2 par le renvoi IF OP$ = T THEN RETURN.
- Le sous-programme 5000 aiguille, dans l'option T, de la boucle 1 à la boucle 2 par le jeu du paramètre N et de l'instruction ON N GOSUB.
- Les boucles 1 et 2 ne sont pas du type FOR NEXT, pour éviter les difficultés des sorties et des rentrées successives dans ce genre de boucle. On sait en effet qu'on ne peut rentrer dans une boucle FOR NEXT que si l'on en est sorti par un GOSUB : c'est le cas pour la boucle 1, mais la boucle 2 ne comprend que des RETURN.
- Il est facile de suivre le programme à la trace à travers les deux boucles

et de constater les différences de modes : dans l'option P les boucles sont parcourues en entier et successivement ; danx l'option T il y a un va-et-vient permanent entre les deux boucles, tranche après tranche.

- N'importe quelle heuristique peut être traitée par ce procédé, le nombre de tranches élémentaires, dont chacune correspond à un temps "mental" pris comme unité [2], étant différent suivant les heuristiques concernées.

2. Chaque tranche devrait, en toute rigueur, comporter le même nombre d'*inférences logiques*, ces opérations qui servent à définir la vitesse d'exécution, en "lips", des ordinateurs.

ÉTAPE E34

QUI DIT MIEUX ?

Présentation

Le programme du présent scénario présente deux missiles autoguidés en compétition, comme en E27. Mais ces missiles sont lourds et possèdent donc, comme dans la réalité, une certaine inertie, contrairement à ceux de E27 qui prenaient instantanément la direction assginée par chacune de leurs deux heuristiques.

Rappelons que le robot ordinaire fonctionnait avec l'heuristique 2 et le robot superintelligent avec l'heuristique 6 à anticipation.

Cette fois, c'est la force de propulsion des missiles (c'est-à-dire leur *accélération*) et non plus directement leur vitesse qui est définie par l'heuristique 2 ou 6.

Les trajectoires sont différentes de celles de E27 car il faut aux missiles du temps et de l'espace pour incurver leur route vers la cible en mouvement.

La formule d'anticipation optimale est difficile à imaginer.

Le programme présenté a adopté la même que celle de E27 mais il doit être possible de faire mieux : telle est l'invitation faite au lecteur.

Comme en E27, seules les deux lignes donnant les composantes AXMIO et AYMIO peuvent être changées, pour assurer la loyauté de la compétition ; les missiles ne doivent pas franchir plus d'une case par pas.

Des effets de bords sont par ailleurs à prévoir pour les deux missiles.

Programme

```
100 'ETAPE E34:COMME E27 AVEC MISSILES "
LOURDS" DOTES D'INERTIE,HEURISTIQUES D'A
CCELERATION (6) ET (2)
110 INK 0,22:INK 1,1:INK 2,6:INK 3,23:CL
S:CLEAR:SCO=0:SCA=0'INIT SCORE
120 SYMBOL AFTER 32:SYMBOL 33,&F0,&E0,&F
0,&B8,&1C,&F,&6,&4
130 CLS:R=0:PEN 1:LOCATE 4,11:PRINT"VITE
SSE DE LA CIBLE (DE 0,5 A 2) ?":LOCATE 2
,13:PRINT"PUIS APPUYEZ SUR LA BARRE D'ES
PACEMENT"
140 D$=INKEY$:RANDOMIZE TIME:IF D$=" " T.
```

```
HEN 150 ELSE IF D$<"." OR D$>"9" THEN SO
UND 1,30,5,7:FOR AT=1 TO 200:NEXT:GOTO 1
40 ELSE R=R+1:A$(R)=D$:PEN 1:LOCATE 17+R
,15:PRINT A$(R):SOUND 1,20,10,7:GOTO 140
150 K$=" ":FOR Z=1 TO R:K$=K$+A$(Z):NEXT
 Z:K=VAL(K$):ON ERROR GOTO 130
160 PEN 1:LOCATE 16,16:PRINT"VIT=";K
170 IF K<0.5 OR K>2 THEN SOUND 1,500,30,
7:FOR AT=1 TO 1000:NEXT:GOTO 130
180 FOR AT=1 TO 1000:NEXT:CLS:H=INT(2*RN
D):IF H=1 THEN PXC=INT(10*RND+8):PYC=6 E
LSE PXC=8:PYC=INT(10*RND+8)'TIRAGE POSIT
ION CIBLE
190 PPXC=PXC:PPYC=PYC
200 PXMI=INT(25*RND)+9:PYMI=INT10*RND+13
:PPXMI=PXMI:PPYMI=PYMI'POSITION DEPART M
ISSILE ET MISE EN RESERVE POUR EFFACEMEN
T
210 PXMIO=PXMI:PYMIO=PYMI:PPXMIO=PXMIO:P
PYMIO=PYMIO'POSITION DEPART MISSILE ORDI
NAIRE IDENTIQUE A CELLE DU PREMIER
220 PEN 2:LOCATE PXC,PYC:PRINT"o":PEN 1:
LOCATE PXMI,PYMI:PRINT CHR$(33):LOCATE 3
,1:PRINT"ORD=";SCO;"
   ";"ANT=";SCA'AFF CIBLE,MISSILES ET SC
ORE
230 VXC=K*SGN(2*RND-1):VYC=K*SGN(2*RND-1
)'TIRAGE COMPOSANTES VITESE CIBLE
240 SOUND 1,200,80,7:FOR AT=1 TO 2000:NE
XT'TOP DEPART
1000 REM BOUCLE DES MOUVEMENTS
1010 FOR M= 1 TO 500
1020 REM CHANGEMENT DE VITESE CIBLE
1030 CH$=INKEY$:IF CH$<>" " THEN 1040 EL
SE 110
1040 REM MOUVMT CIBLE
1050 PXC=PXC+VXC:PYC=PYC+VYC'CALC NOUVEL
LE POS CIBLE
1060 GOSUB 3000' EFFETS DE BORDS
1070 PEN 2:IF CINT(PXC)<>40 OR CINT(PYC)
<>25 THEN LOCATE PPXC,PPYC:PRINT " ":LOC
ATE PXC,PYC:PRINT"o":SOUND 1,100,5,7:PPX
C=PXC:PPYC=PYC ELSE SOUND 1,53,10,7:PXC=
39:VXC=-ABS(VXC):VYC=-ABS(VYC):GOTO 1070
'AFF CIBLE (SAUF POS 40,25 INTERDITE) ET
 MISE EN RES
1080 GOSUB 5000'CIBLE ATTEINTE
1090 REM MOUVMT MISSILE ORDINAIRE,ACCELE
RATION SUIVANT HEUR (2)
1100 AXMIO=SGN(CINT(PXC)-PXMIO):VYMIO=SG
N(CINT(PYC)-PYMIO):
1110 VXMIO=SGN(VXMIO+AXMIO)
1120 VYMIO=SGN(VYMIO+AXMIO)
1130 PXMIO=PXMIO+VXMIO:PYMIO=PYMIO+VYMIO
```

```
1140 IF PPXMIO<>PPXMI THEN 1150 ELSE IF
PPYMIO<>PPYMI THEN 1150  ELSE PEN 1:LOCA
TE PPXMIO,PPYMIO:PRINT CHR$(33):GOTO 116
0
1150 LOCATE PPXMIO,PPYMIO:PRINT" "
1160 PEN 3:LOCATE PXMIO,PYMIO:PRINT CHR$
(33):SOUND 1,30,5,7:PPXMIO=PXMIO:PPYMIO=
PYMIO
1170 REM MOUVMT MISSILE A ANTICIPATION,A
CCELERATION SUIVANT HEURIST (6)
1180 AXMI=SGN(CINT(PXC)-PXMI+VXC*ABS(ABS
(CINT(PYC)-PYMI)-1))'ADJONCTION D'UN TER
ME COMPLEMENTAIRE A L'HEURIST (2) POUR T
ENIR COMPTE DE LA VITESSE DE LA CIBLE
1190 AYMI=SGN(CINT(PYC)-PYMI+VYC*ABS(ABS
(CINT(PXC)-PXMI)-1))'IDEM

1200 VXMI=SGN(VXMI+AXMI)
1210 VYMI=SGN(VYMI+AYMI)
1220 PXMI=PXMI+VXMI:PYMI=PYMI+VYMI
1230 GOSUB 4000'EFFETS DE BORDS
1240 LOCATE PPXMI,PPYMI:PRINT" ":PEN 1:L
OCATE PXMI,PYMI:PRINT CHR$(33):SOUND 1 3
0,1,7:PPXMI=PXMI:PPYMI=PYMI'AFF ET MISE
EN RES
1250 GOSUB 5000'TEST CIBLE ATTEINTE

1260 FOR AT=1 TO 50:NEXT:NEXT M

1270 END
2000 REM ZONE DES SOUS-PROGRAMMES
3000 REM SSPROG EFFETS DE BORDS CIBLE
3010 IF CINT(PXC) >0 THEN 3020 ELSE SOUN
D 1,53,10,7:PXC=1:VXC=-VXC
3020 IF CINT(PXC)< 41 THEN 3030 ELSE SOU
ND 1,53,10,7:PXC=40:VXC=-VXC
3030 IF CINT(PYC)>0 THEN 3040 ELSE SOUND
 1,53,10,7:PYC=1:VYC=-VYC:RETURN
3040 IF CINT(PYC)<26 THEN RETURN ELSE SO
UND 1,53,10,7:PYC=25:VYC=-VYC:RETURN
4000 REM SSPROG EFFETS DE BORDS MISSILE
A ANTICIPATION
4010 IF PXMI>0 THEN 4020 ELSE PXMI=1:GOT
O 4030
4020 IF PXMI<41  THEN 4030 ELSE PXMI=40
4030 IF PYMI>0 THEN 4040 ELSE PYMI=1:RET
URN
4040 IF PYMI<26 THEN RETURN ELSE PYMI=25
:RETURN
5000 REM SSPROG CIBLE ATTEINTE
5010 IF PXMIO<>CINT(PXC) THEN 5020 ELSE
IF PYMIO<>CINT(PYC)  THEN 5020 ELSE PEN
2:LOCATE PXMIO,PYMIO:PRINT CHR$(33):SCO=
SCO+1:GOTO 5030
```

```
5020 IF PXMI<>CINT(PXC)THEN RETURN ELSE
IF PYMI<>CINT(PYC) THEN RETURN ELSE PEN
2:LOCATE PXMI,PYMI:PRINT CHR$(33):SCA=SC
A+1
5030 FOR FOIS=1 TO 4:SOUND 1,27,5,7:FOR
AT=1 TO 100:NEXT:NEXT:FOR AT=1 TO 800:NE
XT:GOTO 180
```

Commentaires techniques

- Avec des missiles lourds il faut prévoir des sous-programmes d'effets de bords, même pour le missile ordinaire, car leur inertie peut les conduire au-delà des limites d'écran. Cela n'a pas été nécessaire dans le programme proposé car les accélérations sont importantes.
- Suivant la masse et la force propulsive des missiles, les composantes AXMI, AYMI, AXMIO et AYMIO peuvent être modifiées en plaçant un coefficient en tête des quatre formules qui donnent ces composantes. Mais le coefficient doit être le même pour les quatre. Cette règle est également valable pour les nouvelles heuristiques que pourrait trouver le lecteur.

ÉTAPE E35

UNE QUESTION FINALE INSIDIEUSE

Ceci n'est pas un scénario, et n'a pas de programme ni de commentaires techniques. Car ce n'est même pas une étape, c'est simplement l'arrivée et la fin du voyage.

Et c'est aussi le moment d'essayer de répondre à l'une des questions les plus mystérieuses, en permanence sous-jacente dans les 34 programmes du livre.

Nous pouvons observer, au terme des exploits de nos petites créatures, que tous les critères énumérés au Chapitre III, sauf deux, ont été progressivement intégrés, à des degrés divers, dans les programmes présentés.

Parlons d'abord du premier, repéré *n*, et gardons le second, objet de la question mystérieuse, pour la fin.

Le critère *n* n'a pas été complètement mis en œuvre, et pour cause.

Le langage BASIC possède certes ses qualités spécifiques : accessibilité, interactivité, qui constituent deux atouts d'initiation considérables, dont on ne prend vraiment conscience qu'en abordant d'autres langages plus professionnels [3].

Mais il est insuffisamment structuré, non récursif, ligoté dans le cheminement linéaire de ses instructions numérotées. Il ne permet pas d'obtenir le découplage des données et des règles de ses programmes, par opposition à la souplesse de la programmation déclarative.

Mais il est remarquable qu'un tel langage d'initiation, inventé voici maintenant de nombreuses années, puisse donner accès à tant de concepts de l'Intelligence Artificielle, grâce à quelques outils puissants dont ses créateurs l'ont doté dès le départ.

Passons maintenant au critère repéré *c* de la liste du Chapitre III : *il y a un but final précis*. C'est anodin, parce que c'est évident, tout programme doit avoir un but, avec ou sans I.A.

Mais il est moins facile de répondre de façon précise à la question suivante... à moins de ne pas se la poser : à quel endroit du programme le but est-il concrétisé, représenté, signifié à l'ordinateur ?

Si l'on examine les 34 programmes, on peut d'abord affirmer que le but (rejoindre le maître ou l'autre robot) ne figure nulle part, si ce n'est dans le sous-programme "maître rejoint" ou "rencontre des deux robots". Encore ne s'agit-il pas de l'indication du but, ni encore moins de directi-

3. *Cf.* P. Lebeux, *Introduction à Ada*, Éditions Sybex.

ves pour l'atteindre, mais simplement d'un constat : le but est ou n'est pas atteint et, dans le premier cas, le parcours est interrompu.

Considérons alors les plus frustres de nos programmes, E1 et E2 ; ils atteignent leur but une fois sur trois, à l'aide de simples rebonds aléatoires qui sont donc, dans ce cas, les seuls artisans de la victoire. Ce sont les instructions correspondantes qui recèlent donc le fameux but.

En E4, c'est clairement l'heuristique de navigation qui joue le même rôle décisif et unique, tout comme en E3 où le cap de lancement en tenait lieu.

Il en est de même en E5.

À partir de E6, étape où les obstacles apparaissent, l'heuristique de dégagement se conjugue avec la précédente pour assigner le but. Rétrospectivement, on peut d'ailleurs considérer les rebonds de E1 et E2 comme des heuristiques de dégagement dans leur version la plus sommaire.

Ainsi, dans tous les cas, la représentation interne du but à l'ordinateur est assurée par l'ensemble des deux heuristiques du programme. Et l'efficacité maximale est obtenue lorsque ces deux heuristiques s'épaulent mutuellement, comme en témoigne la progression des étapes E7 à E13, ou mieux E28. Dans ce dernier cas, les deux incitations au but convergent parfaitement.

En amont, on trouve d'abord les dégagements B, C et D parfaitement neutres de E7, E8 et E9, qui laissent tout le travail intelligent à leur malheureuse heuristique de navigation et se contentent bêtement de se sortir des guêpiers successifs ; puis apparaissent les dégagements E, F et I, de plus en plus coopératifs, G et H n'étant que des neutres un peu plus évolués. D'ailleurs, si I est si efficace, c'est qu'elle exploite elle-même l'heuristique 1 à seconder.

On pourrait très bien imaginer un programme avec deux heuristiques antagonistes. C'est ce qui se produirait en E24 lorsque l'opérateur appuie sur la touche 4 — appelant ainsi l'heuristique de fuite 4 — si l'on adoptait l'heuristique de dégagement F au lieu de C. Il est facile de comprendre que, dans un tel cas, compte tenu de la densité relativement faible des piliers, c'est toujours l'influence de l'heuristique de navigation qui finit par l'emporter ; lorsqu'il n'y a pas trop de danger, les ruades des soldats ne sont pas assez fréquentes pour empêcher le général d'arriver à ses fins...

Pour en revenir à la définition du but, il faut observer qu'en fait un programme comme E7 n'a pas un but unique : faire rejoindre le maître par le robot. Il se propose aussi, pour ne pas dire surtout, de donner à l'observateur le spectacle de ses démarches mentales et de la trajectoire qui en résulte. Et ces buts annexes sont justement l'objet de toutes les autres instructions du programme.

Ainsi, en définitive, le ou les ''buts'' du programme sont disséminés tout au long de celui-ci, ils l'imprègnent totalement et en constituent la substance même : un programme n'est qu'un immense but composite, comportant des buts étapes traduits par la séquence des instructions.

Il en est inévitablement ainsi avec tous les langages et tous les programmes : dans les systèmes experts, par exemple, c'est le moteur d'inférence qui contient les buts du système ; comme chacun a les siens, toujours peu ou prou différents de ceux du voisin, il paraît très douteux que l'on puisse un jour, ainsi que certains chercheurs l'espèrent, bâtir un moteur d'inférence valable pour toutes les applications au nom de l'unité supérieure de la pensée.

Il y faudrait en amont une unité supérieure des buts, donc des besoins, des désirs et des volontés, ni réelle ni souhaitable.

CONCLUSION

Fermons la boucle et dansons la farandole.

La compréhension est une invention artificielle et autoréférentielle de l'intelligence humaine.

Les plantes et les animaux sont exempts de tous risques d'incompréhension, car ils vivent naturellement, sans se targuer de prétendument comprendre ce qu'ils sont ni ce qu'ils font.

Le clivage entre le comportement naturel : *penser pour vivre*, et le comportement humain se situe dès le premier niveau de la contemplation de nous-mêmes, la réflexion révélant ainsi sa vraie nature : une perversion du comportement, *vivre pour penser*.

L'on pourrait soupçonner la Nature de nous avoir dotés là d'une espèce de capuchon cérébral étanche et réflecteur qui nous fait prendre le dedans pour le dehors, le reflet pour la chose. Ce que notre intelligence nous livre en fait de secrets de l'Univers réel, c'est une série de clichés pris à l'aide d'un objectif braqué vers l'arrière.

De réflexions sur les situations en réflexions sur la réflexion des situations, nous sommes la proie de vertigineux aller et retour par le jeu des multiples miroirs déformants dans lesquels nous sommes enfermés.

Nos sens ouvrent bien quelques fenêtres sur l'extérieur, mais les rais qui, ainsi, nous parviennent sont aussitôt engloutis, déviés, piégés dans des prismes obtus, tordus, ou focalisés dans de folles lentilles, et sans cesse projetés sur les écrans mouvants de nos représentations personnelles.

Nous sommes des boîtes noires du monde, dont les incohérents signaux de sortie n'ont aucune corrélation avec la divine simplicité du milieu qui nous baigne et dans lequel notre corps, hors notre tête, se sent si à l'aise.

Mais, n'ayant pas arrêté là son inexorable processus sédimentaire, la

Nature a posé sur cette première carapace neuronique une deuxième couche, puis une troisième..., les êtres les plus intelligents bénéficiant de la plus forte épaisseur. Le système est génial dans sa stupidité ; il suffit d'empiler sans s'occuper de ce qui bouillonne en dessous.

Alors, la partie se complique encore, indéfiniment, car des interactions nouvelles interviennent ; cela se réfracte, cela fuse et filtre, résonne et s'amplifie, frotte et s'échauffe, ou s'éteint.

Des niveaux supplémentaires de pensées se pressent en hordes sauvages et les miroitements initiaux prennent la consistance de reflets du deuxième ordre, puis du troisième...

Éblouis, nous pensons avoir compris que nous sommes destinés à comprendre. L'organe a créé la fonction.

Pour saisir le sens de cette conclusion, il te faut, lecteur, retourner à son début, car la porte ne s'ouvre que sur l'intérieur...

Nous nous retrouverons alors ici dans un instant, goguenards et éternellement voués à des cascades de clins d'œil.

Roméo et Juliette vont de leur côté, heureux d'ignorer leur ignorance et seulement décidés à foncer l'un vers l'autre, bercés par la douce musique de leurs heuristiques, créatures de créature, aussi insaisissables que le reflet d'un arc-en-ciel.

ANNEXE I

TABLEAU DES HEURISTIQUES UTILISÉES PAR LES ROBOTS DANS LE COURS DU LIVRE

	Heuristiques de navigation	*Étapes*
1	Direction des pas liée à la valeur de la pente de la droite joignant le robot à sa cible : si pente inférieure à 50 %, pas horizontal si pente supérieure à 200 %, pas vertical si pente intermédiaire, pas en diagonale	toutes sauf cas particuliers
2	Heuristique 1 simplifiée : si robot et cible sur même ligne, pas horizontal si robot et cible sur même colonne, pas vertical dans autres cas, pas en diagonale	E4, E5, E24 E30, E33
3	Heuristique 1 avec marge de hasard	E15
4	Heuristique 1 en négatif (fuite)	E24
5	Heuristique 2 en négatif	E30, E31
6	Heuristique 2 avec anticipation sur mouvement de la cible	E27
	Heuristiques de dégagement des obstacles	
A	Tentatives uniformes de direction nord-est	E6
B	Tentatives toutes tirées au hasard	E7
C	Première tentative tirée au hasard, puis exploration circulaire dans un sens tiré au hasard	E8, E16, E18, E19, E20, E24
D	Comme C avec interdiction de retour sur position précédente, sauf si c'est la seule position libre	E9, E10, E11
E	Comme D en commençant l'exploration circulaire à partir de l'obstacle	E12
F	Comme E avec exploration dans les deux sens de rotation puis choix de la meilleure position par fonction d'évaluation	E13, E14, E15, E25, E26, E30, E31, E32
G	Double dégagement suivant C	E17, E21, E22, E23
H	Saut d'une case suivant C	E24
I	Heuristique F avec anticipation à profondeur 1	E28, E29

ANNEXE II

RÉPERTOIRE DE TERMES CLÉS UTILISÉS

Terme	Chapitre ou étape	Terme	Chapitre ou étape	Terme	Chapitre ou étape
actif	E16	identification	E1	robotique	III
aléatoire	E1	indéterminisme	E8	sens	IV
algorithme	E4	inférence	III	simulation	IV
anticipation	IX	interactivité	E24	social	E24
apprentissage	VII	isomorphisme	IV	tableau indicé	E20
arbre	V, E28	langage	I	toucher	E1
argument	E9	libre arbitre	E7	tricher	E17
artificielle	III	logarithmique	E5	Turing	E19
banc d'essai	E33	métaconnaissance	III	variable indicée	E2
boucle	E1	navigation	E4	vue	E3
dégagement	E6	odorat	E26		
déterminisme	E6	ouïe	E26		
distance	E13	pas	E1		
environnement	E5	passif	E16		
évaluation	E13	piste d'essai	E30		
faim	E26	prédicat	E9		
flou	E15	programme	I		
fonction logique	E18	proposition	E2		
fonction d'évaluation	E13	qualitatif	I		
Gödel	III	règles	I		
heuristique	E4	représentation	I		

ANNEXE III

FORMULATION GÉNÉRALE DES HEURISTIQUES DU ROBOT

En parcourant les étapes, en particulier E4 et E13, nous avons dégagé le comportement idéal du robot, qui peut être défini d'une manière très simple.

Plaçons-le dans une case d'écran située horizontalement à H cases de celle du maître, et verticalement à V cases de celle de ce maître.

Nous avons vu que, dans ces conditions, la vraie distance que le robot doit franchir, exprimée en nombre minimal de pas, est égale à la plus grande des deux distances H ou V. Appelons R cette distance réelle.

Le comportement idéal du robot consiste donc, pour chacun de ses pas, à faire diminuer R d'une unité. Si par exemple H vaut 20 et V vaut 8, il est nécessaire et suffisant de ramener H à 19, V pouvant prendre quant à elle indifféremment les valeurs 8, 9 ou 7, ce qui laisse au robot un libre arbitre entre trois positions d'égale qualité. C'est ce qui explique l'égale efficacité des trois méthodes rappelées en E4, et il en existe bien d'autres qui les valent.

Le même principe préside à la progression des heuristiques de dégagement et au choix de la fonction d'évaluation de E13. Il permet par ailleurs d'envisager la conception d'une heuristique globale navigation-dégagement consistant, pour chaque pas, à explorer les trois positions équivalentes possibles [1] et à adopter indifféremment l'une de celles qui sont libres. Si aucune n'est libre, on se rabat sur l'une des trois positions pour lesquelles la distance R, faute de diminuer, reste constante, et ainsi de suite.

On peut bien entendu greffer sur cette heuristique une anticipation de profondeur quelconque. Notons enfin qu'elle n'a rien de commun avec l'heuristique globale évoquée au début du Chapitre X, basée sur la vision de l'environnement.

Sur un autre plan, on aura noté que tous les déplacements des robots du livre sont fondés sur une règle constante, le franchissement obligatoire d'une seule case à chaque pas. C'est une démarche naturelle, anthropocentrique, et dont tout esprit créatif doit songer à sortir au plus vite...

Imaginons par exemple que nous sommes les cavaliers, ou les pions, ou les fous, ou les tours, ou la dame du jeu d'échecs, et non plus le roi comme jusqu'à maintenant, et nous pourrons forger toute une gamme de nouvelles heuristiques entièrement différentes et assister à des trajectoires inattendues, faisant intervenir à loisir l'ensemble des concepts présentés : comportements élémentaires, fonctions d'évaluation, apprentissage, anticipation.

Nous verrions ainsi que les intellects du kangourou, du crabe ou du rhinocéros n'ont aucune raison de ressembler à celui des hommes, mais que ceux-ci sont aptes à les simuler tous.

---

1. En prenant garde au fait que, si H et V ne diffèrent que de 1, il n'y a plus que deux bonnes positions possibles, et une seule si H et V sont égales.

BIBLIOGRAPHIE

P. Beaufils et W. Luther : *Algorithmes*, Sybex.
A. Bonnet : *L'Intelligence Artificielle, promesses et réalités*, InterÉditions.
E. Feigenbaum et P. McCorduck : *La Cinquième Génération*, InterÉditions.
D. Hofstadter : *Gödel, Escher, Bach, les Brins d'une guirlande éternelle*, InterÉditions.
M. James : *Introduction à l'Intelligence Artificielle sur micro-ordinateur*, Eyrolles.
P. Lebeux : *Introduction à Ada*, Sybex.
S. Papert : *Jaillissement de l'esprit*, Flammarion.
J. Pitrat : *Textes, Ordinateurs et Compréhension*, Eyrolles.
La Recherche, n° spécial 170, 1985.
Sciences & Avenir, n° spécial 53, 1985.

Achevé d'imprimer par l'imprimerie TRAXIE, 18-20, bd Ménilmontant - 75020 PARIS

Dépôt légal imprimeur - 3e trimestre 1986

Contrairement à certaines idées reçues, il n'est nullement nécessaire de disposer d'un ordinateur spécial, d'un langage ésotérique ni de circonvolutions cervicales exceptionnelles pour pratiquer, dans de sages limites, l'Intelligence Artificielle.
Un bon micro-ordinateur, équipé d'un honnête BASIC, permet, pourvu que les thèmes traités s'y prêtent, une acquisition des principaux concepts de cette prodigieuse discipline.
Tel est l'objectif de ce livre au fil duquel des corps mobiles sont dotés sur l'écran de comportements surprenants les hissant au niveau de robots intelligents.
Mais, le plus étonnant, ce sont parfois les idées que ces petites créatures nous donnent sur les hommes. C'est pourquoi la langue française tient à apparaître dans cet ouvrage comme le plus évolué des langages de l'informatique.

L'AUTEUR

DANIEL LEBIGRE est ingénieur de l'Ecole nationale supérieure de l'Aéronautique et de l'Espace, ingénieur des Arts et Métiers et ingénieur général honoraire de la SNCF, il est également l'auteur de *MO5 Dynamique, Cinématique, Méthode pour la programmation des jeux* paru aux éditions Sybex.

0220 0986 148 F
9 782736 102203

SYBEX

AMSTRAD PROGRAMMES D'INTELLIGENCE ARTIFICIELLE EN BASIC

Document numérisé avec amour par

AMSTRAD CPC

MÉMOIRE ÉCRITE

https://acpc.me/